Copy No.________

Defence Research and Development Canada
Recherche et développement pour la défense Canada

DEFENCE R&D DÉFENSE

Placing Tactical Data into the MIST and LC2IEDM Systems

Anthony W. Isenor

Defence R&D Canada

Technical Memorandum
DRDC Atlantic TM 2003-168
October 2003

Canada

This page intentionally left blank.

Copy No:

Placing Tactical Data into the MIST and LC2IEDM Systems

Anthony W. Isenor

Defence R&D Canada – Atlantic

Technical Memorandum

DRDC Atlantic TM 2003-168

October 2003

Author

Anthony W. Isenor

Anthony W. Isenor

Defence Scientist

Approved by

J.S. Kennedy

J.S. Kennedy

Head, Maritime Information and Combat Systems Section

Approved for release by

Kirk Foster

DRP Chair

Abstract

This document presents a review of the Maritime Information Sharing Technology (MIST) database and the Land Command and Control Information Exchange Data Model (LC2IEDM) in the context of storing sonar contact information. The set-up of each system is described, as well as the software that supports the systems. Tactical data describing a sonar contact is defined and used to load the MIST database. The same data content is then placed into appropriate tables in the Land Command and Control Information Exchange Database (LC2IEDB). Obvious strengths and weaknesses of both systems are described.

Résumé

Le présent document porte sur un examen de la base de données Maritime Information Sharing Technology (MIST) et du modèle de données d'échange d'information de commandement et de contrôle (Terre) (LC2IEDM) dans le contexte du stockage de données de contact sonar. Il décrit la configuration de chacun des systèmes ainsi que le logiciel de soutien de ces systèmes. Les données tactiques qui décrivent un contact sonar sont définies et utilisées pour le chargement dans la base de données MIST. Les mêmes données sont ensuite placées dans des tables appropriées de la base de données d'échange d'information de commandement et de contrôle (Terre) (LC2IEDB). Une description des points forts et des points faibles évidents des deux systèmes est fournie.

This page intentionally left blank.

Executive summary

Background

The evolution of military systems has reached a stage where developers are building a system-of-systems. Networks are also sufficiently mature to make such developments feasible. As a result, developers considering the system-of-systems approach are now confronted with interfacing and establishing communications between these various components.

The role of a central database in such developments is also evolving. From a simple reformatting argument, the concept of a central database becomes advantageous when the number of systems exceeds three. This is because the transfer and manipulation of the data streams requires fewer transformations when a centralized data repository is used as compared to direct system-to-system communications. For this reason, many groups have begun developing or investigating existing database structures for use in the development of the system-of-systems.

Two such database structures are investigated in this report. The Maritime Information Sharing Technology (MIST) is a joint US-UK effort to define contact information relevant for undersea warfare operations. A second development, the Land Command and Control Information Exchange Data Model (LC2IEDM), is a NATO army-based development intended to store all information pertaining to the operational battle space.

These two data models, and the software required for the models, are detailed in this report. Tactical contact data is defined and used to load both the MIST and the LC2IEDB systems.

Principal Results

This investigation provides a foundation understanding of both the MIST and the LC2IEDM structures. The report explains the details of the MIST contact record, including relationships between individual data units within the MIST record.

The LC2IEDM is then presented and tables are described that are used in examples packaged with the Operational Context Exchange Service (OCXS). An attempt is then made to place the tactical contact data into the LC2IEDB. Several weaknesses of both data systems are described, in terms of either data model structure or supporting documentation.

Significance of Results

This investigation documents the insertion of a tactical contact record into the MIST and LC2IEDB structures. Through this investigation, an understanding is built from a contact example rather than documentation that describes the data structures using disjointed examples.

The comparison presented here is important for those considering the use of these databases in future developments. The investigation supports national efforts, such as the Networked Underwater Warfare Technology Demonstration Project currently underway at Defence R&D Canada – Atlantic, as well as international efforts with The Technical Cooperation Program (TTCP). Both groups are exploring central databases for use in the system-of-systems approach.

Future Plans

Efforts within the international community to explore the feasibility of using a central database for communications between systems will continue. The interface mechanism between the LC2IEDB and the integrated systems is being documented. Efforts are also underway to establish replication communications between multiple LC2IEDBs and also extend the data model to incorporate information that is specific to underwater warfare. All of these activities will contribute to the Networked Underwater Warfare Technology Demonstration Project currently underway at Defence R&D Canada – Atlantic.

Isenor, Anthony W. 2003. Placing Tactical Data into the MIST and LC2IEDM Systems, DRDC Atlantic TM 2003-168, Defence R&D Canada – Atlantic.

Sommaire

Contexte

Au stade actuel de l'évolution des systèmes militaires, les développeurs sont en train de construire le système par excellence. Les réseaux également sont suffisamment évolués pour permettre la réalisation des tels développements. Par conséquent, les développeurs qui envisagent d'adopter l'approche basée sur le système par excellence doivent maintenant résoudre le problème de l'interfaçage et de l'établissement des communications entre ces divers éléments.

Le rôle d'une base de données centrale dans ces développements évolue aussi. De simple argument de reformatage, le concept d'une base de données centrale devient un avantage lorsque le nombre de systèmes dépasse trois. En effet, le transfert et la manipulation des flux de données nécessitent un moins grand nombre de transformations lorsqu'on utilise un dépôt de données centralisé plutôt que des communications directes de système à système. C'est pourquoi bon nombre de groupes ont commencé à développer des structures de base de données ou à étudier des structures existantes en vue de les utiliser aux fins du développement du système par excellence.

Deux structures de base de données de ce type sont examinées dans le présent rapport. Le Maritime Information Sharing Technology (MIST) est le résultat d'un projet concerté des É.-U. et du R.-U. visant à définir des données de contact pertinentes pour les opérations de guerre sous-marine. Une autre structure, le modèle de données d'échange d'information de commandement et de contrôle (Terre) (LC2IEDM), est une structure de l'OTAN basée sur l'armée, destinée à permettre le stockage de toutes les données relatives à l'espace de combat opérationnel.

Ces deux modèles de données, et les logiciels qu'ils nécessitent, sont décrits dans le présent rapport. Les données de contact tactiques sont définies et utilisées pour le chargement dans les systèmes MIST et LC2IEDB.

Principaux résultats

Cette étude permet d'acquérir une connaissance de base des structures MIST et LC2IEDM. Le rapport décrit en détail l'enregistrement de contact MIST, y compris les relations entre les unités de données individuelles à l'intérieur de l'enregistrement MIST.

Le rapport décrit ensuite la structure LC2IEDM ainsi que les tables qui sont utilisées dans des exemples intégrés au service d'échange de contexte opérationnel (OCXS). Un essai effectué dans le but de charger les données de contact tactiques dans le système LC2IEDB est ensuite décrit. Plusieurs faiblesses des deux systèmes de

données sont décrites, du point de vue soit de la structure du modèle de données, soit de la documentation à l’appui.

Importance des résultats

La présente étude décrit l’entrée d’un enregistrement de contact tactique dans les structures MIST et LC2IEDB. Par cette étude, on acquiert des connaissances à partir d’un exemple de contact plutôt qu’à partir de documentation décrivant les structures des données à l’aide d’exemples disjoints.

La comparaison présentée est importante pour ceux qui envisagent l’utilisation de ces bases de données dans des développements futurs. L’étude appuie des projets nationaux, comme le projet de démonstration de technologie de guerre sous-marine en réseau actuellement en cours à R & D pour la défense Canada – Atlantique, ainsi que des projets internationaux réalisés avec le Programme de coopération technique (TTCP). Les deux groupes étudient la possibilité d’utiliser des bases de données centrales dans l’approche basée sur le système par excellence.

Plans pour l’avenir

Les activités de la collectivité internationale visant à étudier la possibilité d’utiliser une base de données centrale pour les communications entre les systèmes se poursuivront. On documente actuellement le mécanisme d’interface entre la base LC2IEDB et les systèmes intégrés. Des travaux sont également effectués en vue d’établir des communications de duplication entre plusieurs bases LC2IEDB et en vue d’élargir les modèles de données de façon à intégrer des données qui sont propres à la guerre sous-marine. Toutes ces activités contribueront au projet de démonstration de technologie de guerre sous-marine en réseau actuellement en cours à R & D pour la défense Canada – Atlantique.

Isenor, Anthony W. 2003. Placing Tactical Data into the MIST and LC2IEDM Systems (Entrée de données tactiques dans les systèmes MIST et LC2IEDM), RDDC Atlantique TM 2003-168, R & D pour la défense Canada – Atlantique.

Table of contents

List of figures

List of tables

Acknowledgements

Throughout this investigation, several people were particularly helpful during those moments when suggestions or insight was required. For the MIST system, James Taylor from the Naval Undersea Warfare Centre (NUWC) provided considerable guidance and helpful suggestions regarding MIST and the supporting software. For the LC2IEDM, Frederick Burkley, also from NUWC, provided guidance on OCXS and system set-up. Also regarding LC2IEDM, Eric Dorion (DRDC Valcartier) and LCol Jacques Hamel (DLCSPM) provided considerable help on LC2IEDM structures. Mr. Dorion also provided helpful comments on an initial version of this manuscript.

1. Introduction

Over the years, the conceptual model of the relationship between science and technology has evolved. In the early 20th century, the model perceived the science requirements as driving the technology advancements. A research question would be posed, and technical equipment assembled to address the question. More recently, a conceptual model has emerged where the research-technology relationship is symbiotic, where the questions can originate in either field. In this model, the technology may drive the research [1].

This report is one example of technology driving the research. The military establishment has long considered communications a key to successful operations. Now, in an electronic environment involving multiple national forces, the communication between coalition systems is critical to the efficient leveraging of resources.

Although the conceptual research-technology model is changing, the research procedures within the model have not changed. The typical steps involve the construction or use of systems to collect and analyze the data, data summarization, and then deliberation by the scientist. The procedures are also not unique to military research, but are certainly used in military research. For example, early Canadian research on ship degaussing used these procedures [2]. In degaussing, the changes in the magnetic field were recorded as the ship passed over the range coils. A scientist analyzed the paper recordings and suggested alterations to the ship degaussing coils. In this particular case, the communication of the information (i.e., the needed alterations to the degaussing coils) was in the form of light or flag signals from shore to ship.

These basic procedures of investigation continue to hold regardless of the research-technology model or the present age of electronic processing. In the present scenario, the electronics are collecting the data, the computers are performing data processing, and the scientists are interpreting and then deciding on courses of action. In this scenario, the communications are internal to the processing application or at most, internal to the computer which may be using more than one processing application.

The above scenario describes many systems that have evolved over recent years. However, typically the system or systems have been developed by a single group or more often, a single researcher. As computing became ubiquitous, the necessary application processing was no longer restricted to a single computer. Now, a researcher could distribute the applications and in fact, use applications from other researchers. However, such usage required a complete understanding of the data formats being passed between applications.

Data formats present a considerable obstacle. First, the actual format, or the ordering of the data units, must be fully understood. Second, the content or data itself must be

understood. Through this process of passing data between applications, researchers began to understand that data communications meant more than just transferring data. Data communications implies a level of understanding between the applications.

In today's military research environment, current trends in data communication are toward even greater sharing of data and information. The level of communication is expanding to encompass multiple applications, on different computer platforms, and developed using different software. As well, often the communication involves international applications. In military documents, data and information communication underlies the broad objectives of interoperability and operational integration [3].

In this networked environment, obtaining the data is not the problem. The applications can easily move the data from one computer to another (protocols are quite mature). As well, recent advantages in format structures (e.g., eXtensible Markup Language, XML) have reduced the format-related complications. Now, the problem is extending beyond format issues, to an understanding issue. The receiving applications must be able to transform the format, but also must understand the content. This becomes a more difficult problem as the amount of exchanged data and the number of integrated systems increases.

This transformation and understanding represents an important issue in data transfer between systems. It also represents considerable effort for those involved in system integration. Consider a system-of-systems in its initial stage consisting of only two systems: A and B. In this situation, two conversions are required to transfer data from A to B, and from B to A. As a third system is added, the number of conversions increases to six (e.g., A to B, B to A, A to C, C to A, B to C, C to B). When four systems are linked, 12 conversions are required. In the more general case of n systems, a total of $2*(n!/2!*(n-2)!) = n*(n-1)$ conversions are required (Note: The ! represents the factorial function).

As the concept of a system-of-systems becomes more prevalent, the concept of a central data unit becomes attractive. In this model, a central unit, perhaps a database, acts as a conduit for the data flow. In this model, each application must convert data into and out of the data form used by the central unit.

The central unit addresses both the transfer and context issue. In terms of transfer, the conversion required for a single system into and out of a central unit is two. In the general case of n systems, a total of 2n conversions are required. In this model, linking four systems requires eight format conversions.

However, format conversion is just the first concern. The second is the context or understanding of the data between systems. A central data structure also provides the linked systems with a potential for shared context. In a conceptual sense, shared context implies an understanding and agreement on the meaning of the data units. In a functional sense, this means clear definitions of the data units and relationships between these units.

The military research community is now attempting to create system-of-systems environments. As such, the community is investigating the role of the central database in these environments. In the international community, The Technical Cooperation Program (TTCP), Maritime Systems Technical Panel-One (MAR TP-1), is researching two database systems for possible application to shared data and information in a networked scenario. One database, the Land Command and Control Information Exchange Database (LC2IEDB), originates from the Army Tactical Command and Control Information System program (ATCCIS). A second database, the Maritime Information Sharing Technology (MIST), originates from a collaborative United States (US) – United Kingdom (UK) initiative.

It should be noted that the Land Command and Control Information Exchange Data Model (LC2IEDM) was initially known as the Generic Hub (GH) data model. Many organisations still refer to the model as the Generic Hub. This name originated from the concept of having a generic data model for multipurpose operations that would form a hub for new system development [4]. The name was changed in 1999 to the LC2IEDM to better describe its function. Recent modelling efforts to address joint operations have further evolved the name to Command and Control Information Exchange Data Model (C2IEDM).

Similarly, the MIST system was previously known as the Coalition Data Server (CDS). There will be numerous references to the term “CDS” in command line examples shown in the following report.

1.1 Outline of the Report

This document presents a preliminary investigation involving the LC2IEDB and the MIST systems, and outlines the software used within them. This will provide the reader with a reference frame from which to understand the implementations of the two databases. Next, the implementations used in this study will be given. In reality, each implementation is unique and the exact set-up for this study should not be considered critical. However, details of the implementation provide the language necessary to describe the remainder of the document.

Next, tests for the implementation are presented. These tests are based on the tutorials provided with the software from the originating parties. The tests are useful for understanding both the set-up and functionality of the databases.

Finally, the actual comparison of data flow within the databases is presented. A tactical dataset describing a sonar contact will be defined. This dataset will then be used as the basis for data placement into the MIST and LC2IEDB systems.

The tactical data placement in the databases will illustrate various points. First, it will provide the reader with an understanding of the relationships within both models. The MIST data record is terse, with minimal relationships. This results in quicker understanding for the reader. The LC2IEDB has an extensive table and relationship

structure and will appear complicated to many readers. However, the report will attempt to minimize the complexity by stepping through the data insertion into the LC2IEDB in a progressive manner, documenting the actual data insertion and why it is required for the LC2IEDM structure. Hopefully, this will reduce the complications for the reader.

The second point to consider is the documentation associated with both models. Any insertion of data into a previously unknown data model will test the accompanying documentation. The MIST model, being much smaller, is also reasonably easy to understand. As such, the documentation does not need to be as in-depth because the internal relationships in the model are not extensive. On the other hand, the LC2IEDM has extensive and often complicated relationships. This means the documentation must be capable of presenting these complications to a generally uninformed reader.

Considerable documentation already exists for the LC2IEDM and a report that adds to the documentation is not necessarily a positive point. However, this report is somewhat different from the other LC2IEDM documentation. This report will detail the insertion of a contact record into the LC2IEDM and will explain only those tables used during the insertion. In this regard, this report is application specific, and is not general documentation for the LC2IEDM.

This report also provides many internal links to related parts of the report. Obviously, such links are only useful when examining the electronic version of the report. However, the reader may use the links to progress through the report in a manner appropriate to their needs. For example, someone familiar with the LC2IEDM entities may use the listing of used tables to jump to the contact record content stored in those tables. Conversely, those familiar with the MIST contact record may start at the XML contact document and jump to the LC2IEDM structure being used to store this content.

1.2 Syntax of the Report

Throughout this report, considerable nomenclature will be used to describe the structure of the data models. In the case of the LC2IEDB, *database tables* will be denoted in uppercase characters. The LC2IEDB *column names* will also be uppercase. In the MIST case, the table and column names will be presented in upper camel case, exactly as specified within the database. Note that columns will be associated with tables, and thus the physical implementation of the data model. The logical model will be described using entities and attributes in place of the physical model tables and columns.

The LC2IED system will also be described using both the data model (LC2IEDM) and the database (LC2IEDB). The two (data model and database) are different, as you cannot place data into a data model. The data model is the blueprint for the construction of the database.

XML is used to input data into both the MIST and LC2IE databases. In this report, XML tags will be denoted in the text including the angle brackets <>. This should uniquely identify an XML tag name from a database column name in those cases where the two are the same. The content of the XML tag will be identified using double “quotes” with the actual content in Ariel font.

In some cases, the command line applicable for the execution of programs will be given. In these cases, the command line will be presented in italic text, with the system prompt indicated in regular text.

2. Software Descriptions

There are many software components being utilized within both the MIST and LC2IEDM systems. The databases form only a part of either system. In some cases, similar software components are used in the MIST and the LC2IEDM systems.

The following provides a brief introduction to these software components. The introduction is not intended to convey the details of using or implementing these components. Rather, it is intended to provide an overview of the components.

2.1 Oracle™

Oracle™ [5] is a company specializing in information management software. The Oracle™ product line includes specialized software for database management, development tools, application server tools, collaboration suites and data warehousing tools. Oracle™ is reported to be the second largest software company in the world, with annual revenues of about 10 billion dollars [6] and a database market share of 39.4% [7].

The Oracle™ Company markets the Oracle™ database (hereafter, “Oracle” will refer to the database). The Lite version of the database is available for free download. Personal Oracle9i Release 9.2.0.1.0 (for the Win2000 PC) comes packaged with numerous application tools including the command line interface SQL Plus (SQL represents Structured Query Language). The Oracle database is used in the LC2IEDB system.

SQL Plus provides command line functionality to standard SQL and extended functionality particular to Oracle. The software may be executed either via the windows menu system or from a DOS command window. Some useful SQL Plus commands are listed in Table 1. Documentation for SQL Plus is also available [8].

Table 1. Some commands for the SQL Plus command line interface to an Oracle database.

COMMAND	FUNCTION
help index	Obtain help on SQL Plus commands
desc	List column definitions for a table
quit	Quit the command line interface
@	Allows execution of script file

2.2 PostGreSQL

PostGreSQL is an open source database used in the MIST system. PostGreSQL had its beginnings at the University of California, Berkley, in 1986 [9]. The development of PostGreSQL has since evolved into a Global Development Group, consisting of companies and individuals around the world. Central servers for PostGreSQL are located in Canada.

The PostGreSQL version 7.2 was used in this investigation. A full suite of user manuals provides support for PostGreSQL administrators, programmers, developers, and users [10]. As well, manuals exist for a reference guide and tutorial.

Access to a PostGreSQL database may be obtained via a command line interface. The command line access is executed by typing:

psql [databasename]

The psql command line interface provides very basic SQL interaction with the database including such functionality as SQL Create, Alter, Drop, Insert, etc.

Other important functions that may be executed from the command line are listed in Table 2.

Table 2. Some commands for the psql command line interface to a PostGreSQL database.

COMMAND	FUNCTION
\?	obtain help on psql commands
\h	obtain help on supported SQL commands
\q	quit the psql command line interface
\d	psql command to list all tables in the database
\l	psql command to list all databases

A third party application called Pgaccess [11] may also be used to interact with a PostGreSQL database. This application provides a graphical user interface (GUI) to the database. The GUI provides the ability to create functions, reports, graphs, forms and diagrams of the database tables. These abilities extend the more basic functionality provided by the command line interface.

2.3 Apache Tomcat

The Apache Software Foundation (ASF, formerly known as the Apache Group) is an open source development organization [12] with individual-based membership, not corporation-based. This means that Apache is really a community of developers and users, developing open source software. Apache deals with collaborative and consensus based development, with well-defined membership and by-laws. At present, membership consists of about 90 people.

One Apache project, named Jakarta, has concentrated on open source solutions that run on the Java platform. These Java applications are distributed at no cost. Jakarta consists of many subprojects. At the time of writing, there are 22 subprojects under Jakarta. Tomcat is one sub-project of the Jakarta project.

Tomcat is called a servlet container. This means, in a pragmatic sense, that Tomcat accepts or uses Java code as input. The Tomcat server executes this Java code on behalf of the requesting agent.

The servlet is a special case or type of Java code. A servlet executes on the server. In contrast, an applet executes on the client. Note that executing on the client typically means that the execution is in a browser.

Tomcat functions by listening on a specific port for incoming messages. When a message is received, Tomcat attempts to execute the received message and perform the required action. The normal port for Tomcat is 8080.

Tomcat also handles Java Server Pages (JSP), which is an XML-based script language. The purpose of JSP is to separate the presentation template from the data. JSPs are compiled to create servlets.

2.4 SOAP

The Simple Object Access Protocol (SOAP) is used in the present configuration of the MIST system. SOAP is actually a World Wide Web Consortium (W3C) web services message specification. One implementation of the W3C recommendation has been developed under the Apache Web Services Project [13].

SOAP is described as a lightweight XML-based message protocol used in a distributed environment. In the MIST implementation, SOAP is used as the message protocol to request actions and receive responses from the MIST server. For those unfamiliar with XML-based languages, see [14].

As noted in the previous section, the MIST server provides access to services using Tomcat. Tomcat listens for an incoming request, and when such a request is received, directs the request to the appropriate service. SOAP provides the messaging language

between the client (performing the request) and Tomcat (receiving the request). SOAP is also the messaging protocol from the server back to the client.

A SOAP message consists of three parts: an envelope, the rule for encoding, and the application of the envelope and rules for representing the remote procedure calls (RPC, see Specification [15]). The envelope is a wrapper that describes some of the content and processing associated with the request. The encoding rules describe custom data types. The last section of the message describes the application of the message. This section would contain information on the transport protocol. More information on SOAP may be found in [16].

2.5 ERwin™

ERwin™ is an entity-relationship modelling software package developed by Computer Associates [17]. The LC2IEDM is available in the ERwin™ format, version 3.5.2. The ERwin™ software used during this investigation was version 4.1.2522.

ERwin™ has been used for the full LC2IEDM development cycle. The software provides a graphical user interface to allow the user point-and-click model design capabilities. Entity and relationship properties are quickly accessible. As well, ERwin™ provides a domain dictionary for the creation and management of entities and relationships.

ERwin™ also provides the database design generation. This means that ERwin™ can generate the necessary SQL commands to create the actual database from the model.

There are two important features that make the data modelling software important for this investigation. First, the software allows the visualization of the data model. This is particularly important when attempting to understand the relationships between entities. As well, the user may define the level of complexity in the visualization. This means that areas of the data model may be isolated and visualized without considering the entire model. The LC2IEDM data model diagrams presented in this report are from the ERwin™ software.

Second, ERwin™ allows quick search and selection of entities, attributes, and relationships. This has considerable time saving as compared to searching multiple volumes of documentation to identify all the characteristics of a particular entity, attribute or relationship.

2.6 Operational Context Exchange Service

The Operational Context Exchange Service (OCXS, [18]) is a Java software suite used to access the LC2IEDB. The suite was designed to act as a bridge between the LC2IEDB and outside applications. The suite utilizes the logical and physical XML tag set as based on the logical and physical name sets used in the ERwin™ LC2IEDM.

Data may be input to the LC2IEDB using OCXS and the physical XML name set. If data exist using the logical name set, a converter is provided to transform the logical to physical name sets. This transformation is performed using eXtensible Stylesheet Language Transformations (XSLT). A brief introduction to XSLT is presented in [14].

The OCXS inputs the data to the LC2IEDB using the physical XML name set. All relationships within the LC2IEDB are dealt with implicitly from the data file. OCXS does not deal with any LC2IEDB relationships. This means that the input XML document must load the LC2IEDB tables without violating any formal relationships in the LC2IEDB.

OCXS also deals with output from the LC2IEDB. OCXS can request information from the LC2IEDM, with the information being made available to the user as a physical XML tag set. The OCXS output mechanism was not utilized during this investigation.

3. Hardware and Software Set-up

The hardware / software set-up for this investigation is described in this section. The set-up is slightly different for the two database systems, and thus each system will be described separately. As well, since the MIST system has the added complication of accessing a central database using the SOAP messaging protocol, the protocol will be explained via a simple example.

3.1 LC2IEDB Set-up

The LC2IEDB set-up is shown in Figure 1. The set-up utilizes a single computer, in this case, a Windows 2000 based machine named Jamieson. The supporting software loaded on this computer includes Java, ERwin™, Oracle™, Tomcat, and OCXS.

The Java implementation loaded for the LC2IEDB investigation was jdk1.3.1_04 (Java Development Kit Version 1.3.1_04). The OCXS requires this Java version.

The Oracle installation caused some problems in relation to Tomcat. The Oracle 9i installation automatically assigns port 8080 for the Oracle Hypertext Transfer Protocol (HTTP) Web-based Distributed Authoring and Versioning (WEBDAV) application. However, Tomcat, by default, also wants 8080.

A small revision for the Tomcat server was required to direct Tomcat to port 8079. Under the tomcat/conf directory, two XML files require changes. The two files are: server.xml and test-tomcat.xml. In the server.xml case, a parameter named “port” must be changed to 8079. In the test-tomcat.xml case, a property named “port” must be changed to 8079.

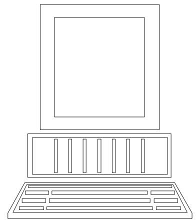

Figure 1. The software implementation for the LC2IEDB investigation. A single Windows 2000 PC named Jamieson is loaded with all required software. The figure also identifies the two primary classes of the OCXS, the Producer and Client classes.

The OCXS does not have a formal installation procedure. Installation consists of unzipping files into a user-defined directory. However, numerous CLASSPATH alterations must be made to allow OCXS access to the proper Java archive (JAR) files. Figure 2 shows the necessary path definitions for OCXS in this implementation.

```
set TOMCAT_HOME=%OCXS_HOME%\Jakarta\build\tomcat
set JAXP_HOME=%OCXS_HOME%\jaxp-1.1

set CLASSPATH=%JAXP_HOME%\jaxp.jar
set CLASSPATH=%JAXP_HOME%\crimson.jar
set CLASSPATH=%JAXP_HOME%\xalan.jar
set CLASSPATH=%ORACLE_HOME%\jdbc\lib\classes12.zip;%CLASSPATH%
set CLASSPATH=%OCXS_HOME%\Ocxs\build;%CLASSPATH%

REM Added by Isenor
SET JAVA_HOME=C:\jdk1.3.1_04
```

Figure 2. Environment variables established for the OCXS use of Tomcat in the LC2IEDB system.

3.2 MIST Set-up

The MIST set-up is shown in Figure 3. This set-up utilizes two computers – a Windows 2000 based client (named Jamieson) and a Linux (version 8) based server (named Maule).

The Windows 2000 client has supporting software including Java and the MIST data server java classes. Note that MIST requires Java version j2sdk1.4.1 (Java Software Development Kit 1.4.1), which is different from the LC2IEDB implementation.

The hardware running Linux has supporting software including Java (j2sdk1.4.0_01), the MIST server Java classes, Tomcat, PostGreSQL, and Pgaccess.

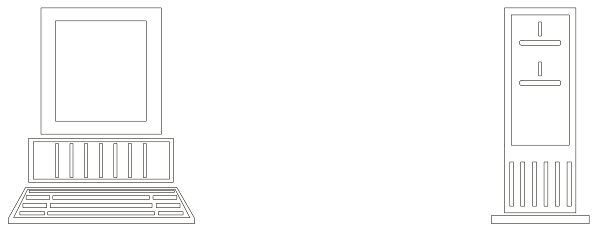

Figure 3. The software implementation for the MIST investigation. A Windows 2000 PC named Jamieson has the role of client, and a Linux box named Maule has the role of server.

3.2.1   SOAP – Tomcat Example

SOAP is the messaging protocol used by the MIST system. Documenting SOAP characteristics is useful for a wider understanding of the MIST system. However, admittedly the protocol details are hidden from the general user. Nevertheless, from a research perspective, the details are worth explanation.

The SOAP protocol and SOAP-Tomcat interaction are best illustrated via a simple example. The example presented here will consist of two pieces of code: the server code or service, and the client code to request the service. This code is not part of the MIST system, but rather is created specifically for this example.

In this example, the service will consist of a simple Java class named DatabaseName (Figure 4). The class contains an integer (dbn) and a string (dbname). The class contains two methods: Db_MIST and Db_LC2IEDM. The Db_MIST method sets the string dbname to “The name is MIST”, while the Db_LC2IEDM method sets the integer dbn to 9. In both cases the set variable is returned.

The DatabaseName class is compiled using j2sdk1.4.1 and placed in a JAR file. The JAR is placed on the server, in this case Maule, in the directory /opt/jakarta-tomcat-4.0.3/webapps/soap/WEB-INF/lib. The service is deployed via the Apache deployment web interface using the parameters shown in Table 3.

The Table 3 parameters allow the SOAP servlet to recognize the specific service and available methods within that service. The ID property defines a name for the service, while the scope property defines how Tomcat loads and maintains the service. The methods property identifies those methods that can be requested. The provider type identifies the code type, and the Java Provider identifies the archive name.

```
public class DatabaseName {
   /** Creates a new instance of DatabaseName */
   int dbn;
   String dbname;
   public DatabaseName(){
      dbn = 0;
      dbname = "";
   }
   public String Db_MIST() {
      dbname = "The Name is MIST";
      return dbname;
   }
   public int Db_LC2IEDM() {
      dbn = 9;
      return dbn;
   }
}
```

Figure 4. A simple Java class in the SOAP-Tomcat service example. This is a service that is placed on the server.

Table 3. SOAP service deployment descriptor for the service example. The properties are described in the text.

PROPERTY	DETAILS
ID	urn:Identify_DB
Scope	Application
Methods	Db_MIST Db_LC2IEDM
Provider Type	Java
Java Provider	DatabaseName

The next step is to produce a client request for the service. The code in Figure 5 provides the request. The code uses the Apache SOAP Java package and RPC. The code establishes the name of the computer that is hosting the service, this being Maule. The Maule port where the Tomcat server is listening is also declared. All requests are first directed to the rpcrouter at this host, via the indicated port number.

The target service and the method name (input as an argument to the class) are also declared. Next, the response is invoked or executed. The return parameters are then obtained and printed to the client screen.

```
import org.apache.soap.*;
import org.apache.soap.rpc.*;
import java.net.*;
public class getName {
   public static void main(String[] args) throws Exception {
      String dbcall = args[0];
      URL url = new URL("http://maule:8080/soap/servlet/rpcrouter");
      Call call = new Call();
      call.setTargetObjectURI("urn:Identify_DB");
      call.setMethodName(dbcall);
      Response resp = call.invoke(url,"");
      Parameter ret = resp.getReturnValue();
      Object value = ret.getValue();
      System.out.println(ret);
      }
   }
```

Figure 5. The client request software for the SOAP-Tomcat example. This code is executed on the client. The code creates a SOAP message that is directed to the server. The message is a request to invoke a service.

Note that the code contains no exception trapping or error reporting. This is the minimal amount of code to perform the service request rather than properly structured code for the reporting of errors.

The Java classes described above represent the software to make the request (Figure 5) and perform the service (Figure 4). When executed, the request generates a SOAP message that is delivered to the server. By utilizing another piece of Apache SOAP software, we can intercept the actual SOAP messages between the client and server.

To intercept the messages, the code presented in Figure 5 requires a slight alteration. The port number 8080 needs to be changed to something else, for example 8070. Then, on the server the TcpTunnelGui (included with the Apache SOAP installation) is initiated by executing:

java -cp /home/CDS/soap.jar org.apache.soap.util.net.TcpTunnelGui 8070 maule 8080

The tunnel software listens for incoming messages on port 8070 and redirects any messages to port 8080. The tunnel software also displays incoming and outgoing messages.

The SOAP message generated by the request is shown in Figure 6. The response message generated by the service is shown in Figure 7.

```
POST /soap/servlet/rpcrouter HTTP/1.0
Host: maule
Content-Type: text/xml; charset=utf-8
Content-Length: 335
SOAPAction: ""

<?xml version='1.0' encoding='UTF-8'?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/1999/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/1999/XMLSchema">
<SOAP-ENV:Body>
<ns1:Db_MIST xmlns:ns1="urn:Identify_DB">
</ns1:Db_MIST>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Figure 6. The SOAP message generated by the client request as shown in Figure 5. The input argument to the class was “Db_MIST”. The SOAP body identifies the requested service in the XML tag set name.

```
HTTP/1.1 200 OK
Content-Type: text/xml; charset=utf-8
Content-Length: 469
Date: Tue, 08 Jul 2003 15:01:38 GMT
Server: Apache Tomcat/4.0.3 (HTTP/1.1 Connector)
Set-Cookie: JSESSIONID=BD21345B8B91910C18AF5EF88FA6A36F;Path=/soap

<?xml version='1.0' encoding='UTF-8'?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/1999/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/1999/XMLSchema">
<SOAP-ENV:Body>
<ns1:Db_MISTResponse xmlns:ns1="urn:Identify_DB" SOAP-
ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
<return xsi:type="xsd:string">The Name is MIST</return>
</ns1:Db_MISTResponse>

</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Figure 7. The SOAP response generated by the server to address the request. The response is based on the code shown in Figure 4.

The client request SOAP message (Figure 6) shows the three sections of the SOAP message as noted in Section 2.4. The namespace established for this message is SOAP-ENV. The SOAP envelope is declared using the XML tag <SOAP-ENV:Envelope>. The Envelope contains a valid XML document. For a very terse description of XML components, see [14].

The Envelope element may contain Header and Body elements. The Header element is optional, while the Body element is mandatory (as per SOAP Specification 1.1, [15]). The Body element encapsulates the actual request or response. In Figure 6, a request is encapsulated within the Body element.

The request within the Body element identifies the method being requested from the service. In this particular example, the request was for the Db_MIST method. Thus, the Db_MIST element is used with the ns1 namespace. This namespace is defined by the attribute in the Db_MIST element. The namespace is described by the service name on the server, in this case urn:Idenitfy_DB (see also Table 3 where the service start parameters are declared).

A brief description of the actual beginning section of the SOAP message is required. This message uses the HTTP POST method as the basis of the transport. Although other transport methods may be used, the SOAP Specification [15] requires

compliance with this particular transport. Next, the /soap/servlet/rpcrouter indicates to the Tomcat server how to direct the incoming request.

The host line indicates where the request is being directed. Since we are on a local network, this request does not need to be explicit. If the request were going across the web, the address might look like www.maule.drdc-rddc.gc.ca. The Content-Type record indicates to the Tomcat server that the message will contain text in the form of XML. Also, the Unicode Transformation Format (UTF) character set is identified. Note that all SOAP 1.1 messages must use text/xml as the content type. The Content-Length indicates the number of characters that follow in the message. This is counted from the end of the Content-Length record. This is because the SOAPAction record represents the first SOAP specific line. SOAPAction is required for all request messages transported using HTTP. The content in the double quotes (in this case blank) helps provide the Tomcat server with the intent of the request. For example, a web address might be included if particular resources are required to meet the request. The blank in this case, indicates that the intent is being provided by the HTTP universal resource identifier (URI), in this case /soap/servlet/rpcrouter.

The SOAP response to the request is shown in Figure 7. The actual XML component looks similar to the request, with the SOAP Envelope and Body elements. However, the response is suffixed to an element that includes the method name, Db_MISTResponse. Again, an attribute declares the namespace used for this new element. The return type used in the code for the Java service method is used to declare a return element. This element is declared as an xsd:string type, with the element content being “The Name is MIST” according to the output from the method.

The beginning of the SOAP response is similar to the request, with HTTP transport being declared, the type and length of the message. However, in the response the date and responding server are also given. The host computer also returns the Set-Cookie record. The cookie identifier is used when a subsequent request requires the same method session as in the original request.

SOAP messaging is implemented in the MIST system to make requests to, and deliver responses from, the central database. Although the methods are different between the above example and the MIST system, the general ideas are the same.

4. Details of the MIST and LC2IEDB Implementation

This section contains very specific information on the LC2IEDB and MIST implementation. The initial part of each subsection is a description of the functionality associated with that subsection. Any specific commands required to meet the functionality are also given.

4.1 LC2IEDB System

4.1.1 Creating the LC2IEDB

The OCXS package comes with an assorted collection of software that allows the creation of the LC2IEDB. The user can create the LC2IEDB by executing an SQL script contained within the OCXS directory structure. The SQL script creates the database tables and all relationships and enumeration lists associated with the LC2IEDB. The input SQL script could be created from the ERwin™ data modelling software, but for this investigation the SQL scripts packaged with the OCXS installation will be used.

To create the database under the Oracle system:

Open a command line window:
Go to directory containing the OCXS installation, in this investigation the Jamieson directory C:\OcxsOct02\Gh5\Gh5DataDefinitionLanguageForOracle. Then enter the commands:

> sqlplus (username:scott password:tiger)
SQL> @creategh5tablespace

This will create the LC2IEDB structure, including all allowable content in the enumeration lists. Note that the @ sign is described in Table 1.

4.1.2 Starting Tomcat

Tomcat must be started on the local machine to permit OCXS communication with the LC2IEDB. To start Tomcat:

Open a command window:
Go to the scripts directory under the OCXS installation. For this investigation, go to directory C:\OcxsOct02\scripts. Then enter the commands to establish the command window environment (see Figure 2) and to start Tomcat:

> *env*
> *starttomcat*

The starttomcat batch file will open a third command line window where Tomcat will be running. Output messages from Tomcat will be directed to this third window.

The Tomcat start-up automatically makes available those methods associated with OCXS. This is because of the directory structure being used by the Tomcat server. In this investigation, the Tomcat server is located on Jamieson at:

C:\OcxsOct02\Jakarta\build\tomcat\webapps

Any directory loaded under the webapps directory on Jamieson, will be loaded as methods available to the Tomcat server. This is the implementation technique used by OCXS.

4.1.3 Test Load of LC2IEDB Using OCXS and Tomcat

As a simple exercise to ensure the database and Tomcat have been successfully deployed, the user should attempt to load the LC2IEDB using the test data provided with the OCXS [18]. In the command line window where Tomcat was started, type:

> *loaddatafill*

The load takes the file C:\OcxsOct02\Ocxs\data\Xformed-OPORD_DATA_FILL.xml and places these data into the LC2IEDB. This XML file contains elements that are defined based on the entities and attributes of the LC2IEDB.

The loaddatafill batch file consists of a command line that directs the content of Xformed-OPORD_DATA_FILL.xml to the LC2IEDB. The LC2IEDB tables used in this load are indicated in Table 4. The definitions of those tables are also given.

The details of the example load will not be considered in this document. However, as explanation, we do consider one small piece of the XML file as shown in Figure 8. Figure 8 shows the XML associated with the OBJ_TYPE table. The OBJ_TYPE table contains information on the type of objects being defined within the LC2IEDB. The Figure shows the seven sub-elements within the <OBJ_TYPE> element. The data insertion uses the exact naming convention as in the LC2IEDM physical data model as defined within the ERwin™ physical model environment. This means that the XML tag names are exactly the LC2IEDB table and column names.

Table 4. Tables used in the loaddatafill example as supplied with the OCXS package [18]. The definitions are based on the LC2IEDM documentation [19] and the ERwin data model.

TABLE NAME	LC2IEDM DEFINITION
REF	An allusion to a source of information that may have military significance.
OBJ_ITEM	An individually identified object that has military significance.
MAT	An OBJECT-ITEM that is equipment, apparatus or supplies of military interest without distinction as to its application for administrative combat purposes.
ORG	An OBJECT-ITEM that is an administrative or functional structure.
UNIT	A military ORGANISATION whose structure is prescribed by competent authority.
ORG_MAT_ASSOC	A relationship of an ORGANISATION as a subject with a materiel as an object.
OBJ_TYPE	An individually identified class of objects that has military significance.
MAT_TYPE	An OBJECT-TYPE that represents equipment, apparatus or supplies of military interest without distinction to its application for administrative or combat purposes.
EQPT_TYPE	A MATERIEL-TYPE that is not intended for consumption.
VESSEL_TYPE	An EQUIPMENT-TYPE that is designed to operate on or under the water surface.
ORG_TYPE	An individually identified class of objects that has military significance.
GOVT_ORG_TYPE	An ORGANISATION-TYPE that controls and administers public policy either under a national or international mandate.
MIL_ORG_TYPE	A GOVERNMENT-ORGANISATION-TYPE that is officially sanctioned and is trained and equipped to exert force.
UNIT_TYPE	A military organisation whose structure is prescribed by competent authority.
MIL_POST_TYPE	A MILITARY-ORGANISATION-TYPE with a set of duties that can be fulfilled by one person.
RPTD	The specification of source, quality and timing that applies to reported data.
RPTD_ABS_TIMING	A REPORTING-DATA that specifies effective date and time that are referenced to Universal Time.
OBJ_ITEM_TYPE	A record of the perceived classification of a specific OBJECT-ITEM as a specific OBJECT-TYPE.
CONTXT	A reference to one or more REPORTING-DATAs.

```
<OBJ_TYPE>
  <OBJ_TYPE_ID>3800000000</OBJ_TYPE_ID>
  <CAT_CODE>MA</CAT_CODE>
  <DUMMY_IND_CODE>NO</DUMMY_IND_CODE>
  <NAME>Sub</NAME>
  <NATIONALITY_CODE>UK</NATIONALITY_CODE>
  <OWNER_ID>1</OWNER_ID>
  <UPDATE_SEQNR>1</UPDATE_SEQNR>
</OBJ_TYPE>
```

Figure 8. Small section of the XML file used to load the LC2IEDB with test data. The data fill inserts a single record into the OBJ_TYPE table.

To clearly show the tag name – column name relationship, we query the LC2IEDB for information regarding the OBJ_TYPE table. Go the SQL query command window to query the system for information on the structure of the OBJ_TYPE table by entering:

SQL> desc obj_type

This command will provide the information given in Figure 9. Note that this is the structure of the table, not the content. The column names in Figure 9 are identical to the XML element names in Figure 8. Also, all attributes in the table are declared as NOT NULL. This indicates that all columns must contain content otherwise the database will generate an SQL fault. The description indicates which columns contain numbers (denoted NUMBER), and those that contain character text (denoted CHAR). The maximum number of allowed characters is also indicated.

OBJ_TYPE_ID	NOT NULL NUMBER(15)
CAT_CODE	NOT NULL CHAR(6)
DUMMY_IND_CODE	NOT NULL CHAR(6)
NAME	NOT NULL VARCHAR2(100)
NATIONALITY_CODE	NOT NULL CHAR(6)
OWNER_ID	NOT NULL NUMBER(11)
UPDATE_SEQNR	NOT NULL NUMBER(15)

Figure 9. The characteristics of the OBJ_TYPE table in the LC2IEDB.

To obtain the content of the table, rather than the structure of the table, enter:

*SQL> select * from obj_type;*

This produces the output shown in Figure 10. The content from the XML file (Figure 8) is evident in the table record.

```
OBJ_TYPE_ID CAT_CO DUMMY_ NAME    NATION   OWNER_ID UPDATE_SEQNR

----------- ------ ------ ------- ------ ---------- ------------

 3800000000 MA     NO     Sub     UK              1            1
```

Figure 10. The content of the OBJ_TYPE table in the LC2IEDB after execution of the loaddatafill.

If all these steps have been successfully completed, then you are assured the LC2IEDB, OCXS and Tomcat are operational on your system.

To remove all content from the LC2IEDB, type

SQL> @DropAndThenCreateGh5TableSpace.sql

This actually removes all content, removes all tables, and then recreates the table structure in the LC2IEDB. Successful completion ensures the database is empty.

4.2 MIST

4.2.1 Starting PostGreSQL for MIST

If installed properly, the PostGreSQL service should have been started when the host machine, in this case Maule, is booted. The PostGreSQL service, called postmaster, should be started according to the directions in [20].

4.2.2 Starting Tomcat for MIST

Tomcat is also utilized in the MIST system. However, for MIST it is executed on the database host machine, in this investigation the server Maule.

To start Tomcat in this implementation, the user must log into the server using the root username and password. Once logged in, typing the following at a command prompt will start Tomcat (again, see [20]):

/opt/jakarta-tomcat-4.0.3/bin/startup.sh

4.2.3 Deploying a Service Under Tomcat for MIST

Once Tomcat is started, services must be deployed for Tomcat. For the LC2IEDB system, the services under Tomcat were deployed automatically. This was because the services were made available under the webapps directory. In the MIST case, the services are started under the administration of SOAP.

To start a service, the user may log in as a normal user, (e.g., on Maule log into the cdsop account), and then open a browser window with the browser directed to:

http://localhost:8080/soap/index.html

A SOAP Administration page should be shown. Click on “Run” Admin Client. This should start the urn:DataServer method for the MIST database.

The urn:DataServer specification should be automatically loaded from the DeployedServices.ds file which should exist under the Tomcat home directory. In this investigation, it is located at /opt/jakarta-tomcat-4.0.3/webapps/soap. However, the urn:DataServer may also be deployed using the Deploy button on the SOAP Administration page. To deploy the urn:DataServer using this method, follow the deployment information provided in Table 5.

Table 5. SOAP service deployment descriptor for the MIST data server. The SOAP properties have been described in Section 3.2.1.

PROPERTY	DETAILS
ID	urn:DataServer
Scope	Application
Methods	addContact getContact addContactList getAllContacts removeContact
Provider Type	Java
Java Provider	usuk.ds.DS

4.2.4 Initializing the MIST Database

The MIST database needs to be created and initialized with the appropriate table structure. This procedure is conducted on the server machine, in this case Maule. The PostGreSQL postmaster service must be running on the server to execute these commands.

The database is created and initialized by executing the commands:

% *createdb CDS*

% *initdb /home/CDS*

where CDS is used as the database name. This procedure is outlined in [20].

Next, the main table in the database is created. From the client machine, in this case Jamieson, the InitializeDB class is executed [20]:

java -cp /home/CDS/jdbc7.0-1.2.jar:/home/CDS/CoalitionDS.jar:/home/CDS/soap.jar:/home/CDS/activation.jar:/home/CDS/mail.jar usuk.ds.util.InitializeDB

This creates the table "ContactTable" in the MIST database. The ContactTable maintains a list of all active contacts in the MIST system. The table contains two columns: "Name" and "Reports". The Name column identifies the name of a particular contact. The Reports column describes how many reports are available for that particular contact name.

The ContactTable Name column also identifies the existence of another table in the MIST database. Every contact named in the Name column will have an associated table with the same name in the MIST system. The number of records in the contact named table, will be identified in the ContactTable Report column.

4.2.5 Test Load of the MIST Database

To test the creation and communication to the MIST database, the user should add a contact to the database. To complete this test, open a command window on the client computer, (in this case Jamieson). If Oracle has been installed on the client (in this investigation it has been installed for the LC2IEDB system), move to the C:\CDS directory and set the environment path as follows:

SET PATH=C:\WINNT;C:\WINNT\SYSTEM32

The Oracle installation alters the Java runtime environment, requiring the resetting of the PATH variable.

After starting Tomcat on the server (see Section 4.2.2), starting the SOAP service (Section 4.2.3), and creating and initializing the database (Section 4.2.4), the test insertion may be performed from the client command window using the following command:

java -cp soap.jar;activation.jar;mail.jar;CoalitionDS.jar;. usuk.example.AddContact http://maule:8080/soap/servlet/rpcrouter Test1

As verification of the successful completion of the command, the user may open a Pgaccess window on the server (in this case, Maule). Opening the database, the user should see the tables “ContactTable” and “Test1”. Not that Test1 is the table name defined in the command line given above.

The Java class AddContact is part of the tutorial packaged with the MIST installation [21]. The actual contact data is generated in the AddContact class.

5. The Contact Data

One purpose of this report is to document the loading of contact data into the two database systems. For this investigation, the data content of a tactical sonar contact is assumed to consist of the information presented in Table 6. Essentially, the content identifies the owner of the contact data, space-time characteristics of the contact (e.g., position, course, speed, etc.) and the identifying characteristics of the contact (e.g., domain, type, identifiers, etc.).

For this investigation, the contact position will be described as a range and bearing from the ownship position. This alteration helps maximize the input data content, thereby exercising more components of each database. The data content used to load each data system is shown in Table 7.

The data content (Table 7) indicates that HMCS Grove has identified a contact. The contact is at a range of 12 ± 0.5 km and a bearing of $270 \pm 6.75°$. The contact is travelling due east (i.e., approaching HMCS Grove) at a speed of 5 m/s. The contact is a submarine and is operating at a depth of 100 ± 10 m. The submarine is known to be hostile and has various markings that are also known.

In the work that follows, it will be natural for the reader to begin comparing the two databases being considered. However, **it should be stressed that comparing databases is inherently unfair**. That is because databases are designed to fulfill a particular need. If the data coming into the database is not associated with the initial database requirement, then the data will not find a logical home within the database structure. Here, we assess the databases ability to store contact data, and also assess the database documentation.

Table 6. Defining the data characteristics of the contact information.

CHARACTERISTIC	DEFINITION
Owner	Owner of the data. Note that owner is considered the platform where the sensor is based that took the measurements that identified the contact.
Sensor Type and Frequency	Type of sonar sensor including the frequency of operation.
Contact Position	Specific point in x,y,z space where the contact is located.
Time	Time the contact position was determined.
Position Error	A measure of error associated with the contacts position.
Contact Course	The course of the contact.
Contact Speed	The speed of the contact.
Track Number	Track number of the contact
Components	Component tracks that may have been fused to create the track.
Country of Origin	The contacts country of origin.
Domain of Operation	The spatial domain in which the contact is capable of operations.
Type of Platform	The type of vessel the contact is thought to be.
Identifiers	Any unique identifiers for the contact.
Threat Level	The threat level posed by the contact to the Owner.
Security Classification	The security classification of this contact data.

Table 7. Data used to fill the MIST and LC2 systems. In some cases, the data are unrealistic (e.g., depth uncertainty of 20 m). However, the exact data value is not important. The important point is determining whether or not the database can store the described data.

CHARACTERISTIC	DATA CONTENT
Owner	HMCS Grove
Sensor Type and Frequency	Sonar sensor operating at 300 Hz
Reporting Platform position	45 N, 50 W
Time	June 24, 2003 at 11:10:00
Contact Course	Due east (90 degrees true)
Contact Speed	5 m/s (=18 km/hr)
Other contact information	Contact is at a bearing of 270 degrees from the reporting platform, and a range of 12 km. Bearing uncertainty is 13.5 degrees, while range uncertainty is 1 km. The depth of the contact is 100 m below the water surface. The uncertainty in depth is 20 m.
Track Number	BAD001
Components	This track is a fusion of tracks 1 and 4.
Country of Origin	Thought to be a unit from Bosnia and Herzegovina.
Domain of Operation	Underwater
Type of Platform	Submarine
Identifiers	Known to be a Model 1234 submarine, orange in colour with a large yellow dot.
Threat Level	This is a hostile submarine.
Security Classification	The contact information should be shared with only the Canadian partners. US and UK partners know positions of Canadian assets.

6. Placing Contact Information into MIST

The following section will describe the placement of the contact information into the MIST database. The data load will be described as well as issues resulting from the data load.

6.1 MIST Modifications

The MIST system was modified slightly for this assessment. The Java code for the AddContact class [21] was used as a starting point for the construction of the AddConact2 class. AddContact2 reads an XML input file in the form of a MIST contact record. The content of the XML document is then placed into the MIST database. This modification separates the process of creating the XML document from the loading of the content into the database.

6.2 Loading the MIST Database

The data content shown in Table 7 is now placed in the MIST database. Since the database contains only two tables (only one table containing contact data, see Section 4.2.4), the table structure will not be shown. Rather, the data placement in the XML MIST record will be shown. The XML record is chosen over the table, because the XML actually contains internal structure that may help the reader group data units.

The MIST contact data used in this report is shown in Figure 11. The structure consists of three sections, which are described here.

The first section describes two characteristics of the sensor used to identify the contact. The sensor used to identify the contact is of <TYPE> "Sonar", as per the data shown in Table 7. The section also lists the frequency used by the sensor, in this case "300" Hz [21].

The second section deals with the position of the contact. The position characteristics within the MIST model are variable, depending on the specification of the <Bearing>/<BearingError> pair. When the <Bearing>/<BearingError> pair are used in the contact specification, then the <Latitude>/<Longitude> pair represents the position of the reporting platform. If the <Bearing>/<BearingError> are not used, then the <Latitude>/<Longitude> pair indicates the position of the contact [21]. In this investigation we choose to reference the contact relative to the position of ownship. Thus, we include the <Bearing> information in the MIST XML document.

```xml
<?xml version="1.0"?>
<!--MIST test_contact_1 XML file.
Uses AddContact2.java which I have packaged inside
CoalitionDS.jar
       Anthony W. Isenor April 2003-->
<Contact>
  <OwnershipID>HMCS Grove</OwnershipID>
  <Sensor>
    <Type>Sonar</Type>
    <SonarFrequencies>300</SonarFrequencies>
  </Sensor>
  <Position>
    <Latitude>45</Latitude>
    <Longitude>-50</Longitude>
    <Altitude>-100</Altitude>
    <PositionError>10</PositionError>
    <Course>90</Course>
    <Speed>5</Speed>
    <Bearing>270</Bearing>
    <BearingError>5</BearingError>
    <Range>12000</Range>
    <RangeError>8.33</RangeError>
  </Position>
  <Identification>
    <CoalitionTrackNumber>BAD001</CoalitionTrackNumber>
    <Composites>1,4</Composites>
    <Country>BA</Country>
    <Domain>Subsea</Domain>
    <Type>Sub</Type>
    <Specific>Model 1234</Specific>
    <UniqueID>Orange with big yellow dot</UniqueID>
    <ThreatLevel>Hostile</ThreatLevel>
  </Identification>
  <TimeStamp>1056463800000</TimeStamp>
  <SecurityClassification>CA</SecurityClassification>
  <UnRestrictedReleaseList>US,UK</UnRestrictedReleaseList>
</Contact>
```

Figure 11. The MIST XML-based contact content. Note that altitude error cannot be stored in the MIST record structure. Links are provided in the content (indicated by bold characters) to those sections of the report where the information is loaded into the LC2IEDB. Links are also provided in the tag names. Clicking on a tag name will reposition the text to the report section explaining issues regarding the particular tag name.

The other characteristics of the position include the <Altitude>, <Course>, <Speed>, <Range> and <RangeError>. All of these characteristics refer to the contact. The

<PositionError> refers to the error on the <Latitude>/<Longitude> pair. This characteristic may refer to the contact or the reporting platform, as per the above description.

The specific data content is defined as per Table 7. The position of the reporting platform is 45°N, -50°E. The contact is reported to be travelling in a direction of 90°(<Course>), at a speed of 5 m/s (<Speed>). Relative to the Ownship, the contact is at a <Bearing> of 270°, and a <Range> of 12000 metres. The contact is also reported to be at an <Altitude> of 100 m below the surface.

The third section lists characteristics to identify the contact. All information content in the XML Identification tag pertains to the contact. This sections lists:

- <CoalitionTrackNumber>,
- <Composites> – used to determine the track,
- <Country> – origin of the contact (if known)
- <Domain> – sea, air, undersea,
- <Type> – a generic type of platform represented by the contact
- <Specific> – additional information regarding the contact
- <UniqueId> – unique identifiers for the contact
- <ThreatLevel> – assumed threat of the contact

In the XML document (Figure 11), the contact track number is identified as "BAD001", and is based on <Composites> "1,4". The contact is from <Country> "BA" as defined by the ISO 3166 country codes [22]. This code list identifies BA as Bosnia and Herzegovina (Note: the MIST documentation refers to 3-character ISO 3166 country codes, when in fact the code base is 2-character). In this contact record the domain is identified as "Subsea". The contact is described as type "Sub", "Model 1234" and has some unique colours ("Orange with a big yellow dot"). It is also considered hostile.

The fourth section of the XML document deals with miscellaneous information regarding the contact. For example, the time of the contact position, which is specified in milliseconds from 00:00 January 1, 1970. The security information is also present in this section. The <SecurityClassification> identifies the allowed distribution list of countries for the contact information. Finally, the <UnRestrictedReleaseList> identifies those countries that are permitted to receive Ownship's position. For both <SecurityClassification> and <UnRestrictedReleaseList>, the ISO 3166 country codes are used [21].

6.3 Issues Related to Information Placement into the MIST

The following section describes issues relating to the insertion of the contact data into the MIST database.

6.3.1 Range and Bearing Error

In the MIST system, the errors associated with the position, bearing and range are all expressed as the percentage of error associated with the measure. For example, if the error tag contains a “5”, then the reporting unit is 95% certain that the contact is at the stated position, range or bearing [21]. This seems to imply the percentage certainty in occupying a specific location in space, rather than the spread of a distribution encompassing a specific location.

The contact information used in this example has a range error of 1 km. Given this particular definition of range error, it is not clear how this 1 km translates to a percentage range error. Similar confusion exists for bearing and position error.

For this investigation, we assumed a simple relationship between the uncertainty and range. Given an error of 1 km, the percentage error is assumed to be 1 km/ 12 km * 100% = 8.33%.

Bearing error introduces another problem. If we treat bearing error similar to range error, we are unable to specify a non-zero error when the bearing is zero degrees.

6.3.2 PositionError

The position error is included in the XML document shown in Figure 11 for completeness. However, when the MIST record contains bearing information, the position information pertains to Ownship. In this case, the <PositionError> tag is somewhat meaningless. When bearing information is not included in the MIST record, the position information refers to the contact position. In this case, the definition of <PositionError> becomes important.

The <PositionError> is defined similar to the <RangeError> and <BearingError>. The definition is “the percentage of error associated with this contact’s position (i.e., 5.0 is you are 95% sure it is at that given location)”[20]. However, the MIST documentation also refers to “the uncertainty area described by <RangeError> and <BearingError>” (see [20] in the description of <UnRestrictedReleaseList>). These conflicting definitions weaken the MIST documentation.

6.3.3 Altitude

The contact information notes that the altitude of the contact has an associated uncertainty of 20 m. There is currently no location within the MIST database for storing this information.

6.3.4 Latitude and Longitude

The <Latitude>/<Longitude> tag implementation in the MIST XML document presents structural problems. The reason deals with the <Latitude>/<Longitude> pair and the <Bearing>. When the <Bearing> element is used, the <Latitude>/<Longitude> pair describes the position of the contact. When the <Bearing> element is not present, the <Latitude>/<Longitude> pair describes the position of the Ownship. In the Ownship case, including the <PositionError> is somewhat meaningless, as the Ownship position should be known.

This dual purpose for these tags may introduce confusion. The structure could be revised to encapsulate the <Latitude>/<Longitude> tags into an element that clearly identifies the referenced platform (OwnshipID or the contact).

6.3.5 Comma Separators

Comma separators within XML content (e.g., <Composites>) are not illegal, but there are implications. Standard XML parsers will not be able to parse out the individual content in the string. This may or may not be important, but the implication should be acknowledged.

6.3.6 XML Structure and the Database

The difference in structure between the XML MIST record and the MIST database should be noted. Although it is not essential that both the XML and database contain the same structure, differences may introduce confusion for some readers.

7. Placing Contact Information into LC2IEDB

In the LC2IEDB system, the organization of the contact information is not trivial. As such, the following section will describe the LC2IEDB data insertion in detail. The description will follow the organisation inherent within the LC2IEDM structure.

7.1 LC2IEDB/OCXS Modifications

The only modification to the LC2IEDB system involved the OCXS eXtensible Sytlesheet Language Transformations (XSLT) translator. The transformation was expanded to include transformations from logical to physical XML for the following entities:

- Electronic-Equipment-Type
- Capability
- Mobility-Capability
- Object-Item-Capability
- Coordinate-System
- Relative-Point
- Object-Reference
- Context-Element
- Geometric-Volume
- Surface-Volume

7.2 Contact Data Load into LC2IEDB

The load into the LC2IEDB involves two steps. Although this is not necessary, the data content was first described in what has been referred to as verbose XML. It is then converted, using an XSLT conversion named verboseXMLtoterseDB.xsl, to the terse tag set. More correctly, these two tag sets represent the logical (verbose) and physical (terse) tags used in the data model. The ERwin™ logical and physical data

model views have the corresponding attributes and column names, respectively. In this document, the two tag sets will be referred to as logical and physical.

Creating the input in the logical XML provides one key advantage as compared to the physical XML. Working in the logical structure means you can relegate structure issues to the XSLT. This is convenient as the relationships within the LC2IEDB place considerable restrictions on the data loading. This also has the result of allowing easier insertions and deletions from the content, again because the XML writer may ignore the details of the structure.

The physical tag set will be considered in this document. This is because the logical XML does not account for the order dependencies that are mandatory in the physical XML. These order dependencies are a result of the relationships established in the physical model. The XSLT conversion takes into account the relationships, and thus creates a physical XML structure that meets the loading order required as a result of the data model relationships.

The physical tag set XML is loaded into the LC2IEDB using the OCXS. The load is executed on the client using the OCXS Producer code according to the following command: (Note: this is one continuous command line):

java ocxs.client.OcxsProducerClient -s
http://localhost:8079/ocxs/OcxsProducerServlet -x
"%OCXS_HOME%\Ocxs\data\Contact_Physical.xml"

The physical model XML will also be presented in the sections that follow. The XML is presented in the order that meets the relationship requirements of the LC2IEDB. A contiguous and complete physical XML document is presented in Annex 1. Note that the order of the XML in the Annex will not be the same as the order presented sequentially in the figures. This is because the figures present the XML in an order that groups related information. However, the XML in the figures will still meet the relationship requirements of the LC2IEDM, if used in the figure order.

The physical XML will utilise the tables listed in Table 8. The order of the tables presented in Table 8 is consistent with the order of the XML snippets that follow in the figures. The table names in Table 8 are also linked to the figures containing the XML snippets that load the content. This provides those LC2IEDM familiar readers using the electronic version with quick access to the tables used in this investigation.

Table 8. The tables of the LC2IEDB used in the insertion of the contact record. The left column indicates the table name used in the LC2IEDB. In the electronic version of this document, these names are linked to the Figure that uses the particular table. The definitions are directly from the LC2IEDM ERwin documentation for Draft 5.3 dated 18 Feb. 2003.

TABLE NAME	LC2IEDM DEFINITION
REF	An allusion to a source of information that may have military significance.
OBJ_ITEM	An individually identified object that has military significance
MAT	An OBJECT-ITEM that is equipment, apparatus or supplies of military interest without distinction as to its application for administrative combat purposes.
ORG	An OBJECT-ITEM that is an administrative or functional structure.
OBJ_TYPE	An individually identified class of objects that has military significance.
MAT_TYPE	An OBJECT-TYPE that represents equipment, apparatus or supplies of military interest without distinction to its application for administrative or combat purposes.
EQPT_TYPE	A MATERIEL-TYPE that is not intended for consumption.
ELCTRNC_EQPT	An EQUIPMENT-TYPE that is designed to use electronic processing to realise its primary function.
RPTD	The specification of source, quality and timing that applies to reported data.
RPTD_ABS_TIMING	A REPORTING-DATA that specifies effective date and time that are referenced to Universal Time.
OBJ_ITEM_STAT	A record of the perceived condition of a specific OBJECT-ITEM as determined by the reporting organisation.
LOC	A specification of position and geometry with respect to a specified horizontal frame of reference and a vertical distance measured from a specified datum.
POINT	A zero dimensional LOCATION.
ABS_POINT	A POINT that has its coordinates specified with respect to a given horizontal frame of reference and may have a vertical distance specified.
VER_DIST	A specification of the altitude or height of a point or a level as measured with respect to a specified reference datum in the direction normal to the plane that is tangent to the WGS84 ellipsoid of revolution.
SURFACE	A two-dimensional LOCATION.
FAN_AREA	A SURFACE that is in the form of a truncated ring sector, which is a sector lying between and being bounded by the rays emanating from the central point of the ring and having a specified central angle.
GEOM_VOL	A specific LOCATION that is a three-dimensional bounded space.

SURFACE_VOL	A GEOMETRIC-VOLUME that has its horizontal boundaries defined by a specific SURFACE.
OBJ_ITEM_LOC	An association of an OBJECT-ITEM with a LOCATION that enables the geographic position of the OBJECT-ITEM to be specified. The specification may include spatial characteristics to which operational meaning is attributed in terms of a type of control feature.
CAPAB	The potential ability to do work, perform a function or mission, achieve an objective, or provide a service.
MOB_CAPAB	A CAPABILITY, required for planning, of those FACILITYs, MATERIELs, ORGANISATIONs and PERSONs or FACILITY-TYPEs, EQUIPMENT-TYPEs, ORGANISATION-TYPEs and PERSON-TYPEs that are deemed as having the nominal ability to move in the air, on water, under water, or over a specific type of terrain.
OBJ_ITEM_CAPAB	A perceived value of a specific CAPABILITY of an OBJECT-ITEM.
CONTXT	A reference to one or more REPORTING-DATAs.
CONTXT_RPTD_ASSOC	A relationship of a CONTEXT as a subject and a REPORTING-DATA as an object.
CONTXT_ELMT	A reference to a specific REPORTING-DATA that is a constituent part of a specific CONTEXT.

The first section of the physical XML is shown in Figure 12. Since this is the first XML section to be described, the XML version tag begins the text, followed by the <GH5Complete> tag and namespace attribute. Next, the reference (REF) table is identified and a single record in this table is declared. Note that the <REF_TBL> tag identifies the table to be loaded, while the <REF> tag identifies a record for the table. This form of table and record representation is used throughout the XML that is required to load the LC2IEDB using OCXS.

The REF table provides reference to a source of information. The <REF_ID> uniquely identifies a source, while the <DESCR_TXT> describes a particular reference. The <SOURCE_TXT> identifies the originator of the source information. The <FORMAT_CODE> is used to identify if a prescribed format exists for the reference. The "NOS" indicates no prescribed format. The <SECURITY_CLSFC_CODE> indicates a North Atlantic Treaty Organisation (NATO) security classification that pertains to the reference. In this case, "NU" indicates NATO Unclassified. The <TRANS_TYPE_CODE> indicates the mechanism by which the reference is transmitted to the recipient. In this case, the "EMLMSG" represents a message received through an email system.

```xml
<?xml version='1.0' ?>
<GH5Complete xmlns:dt="urn:schemas-microsoft-com:datatypes">
 <REF_TBL>
  <REF>
      <REF_ID>18</REF_ID>
      <FORMAT_CODE>NOS</FORMAT_CODE>
      <DESCR_TXT>Canadian Source</DESCR_TXT>
      <SECURITY_CLSFC_CODE>NU</SECURITY_CLSFC_CODE>
      <SOURCE_TXT>Grove Contact Information</SOURCE_TXT>
      <TRANS_TYPE_CODE>EMLMSG</TRANS_TYPE_CODE>
      <OWNER_ID>7</OWNER_ID>
      <UPDATE_SEQNR>1</UPDATE_SEQNR>
  </REF>
 </REF_TBL>
```

Figure 12. The first section of the physical XML establishes the reference. The reference provides a terse description of the information source.

The <OWN_ID> and <UPDATE_SEQNR> tags are somewhat special to the data model. These particular tags exist in all tables in the LC2IEDB. These columns help control the replication management between LC2IEDB instances. The <OWNER_ID> identifies the owner of a data item or record in a table. The <UPDATE_SEQNR> is an absolute sequence number that determines the seniority of a data item. The exact role of these columns in the replication management is being investigated, but will not be explored further in this report. Throughout this report, the <OWNER_ID> will be described as "7", while the <UPDATE_SEQNR> will be described as "1".

The second section of the physical XML deals with the object items described in the LC2IEDM. Object items are used to describe general items that may be used throughout the data model. In this example, two object items need to be described. The first is the contact, which in this example is an enemy submarine. The second is our operating platform, which is a vessel. These object items are described using two records in the OBJ_ITEM table (Figure 13). The first record assigns the unique code of "28" to the item. The item is categorized <CAT_CODE> as a Materiel (denoted with code "MA") and is given the <NAME> "BAD Guy 1". The item is also given an alternate identification of "BAD001".

The <OBJ_ITEM> record for Ownship is also shown in Figure 13. The important component of this record is the unique identifier, which in this case is assigned a value of "10". The Ownship name is given as "HMCS Grove" and an alternate identification is assigned "CAD001".

```
<OBJ_ITEM_TBL>
  <OBJ_ITEM>
    <OBJ_ITEM_ID>28</OBJ_ITEM_ID>
    <CAT_CODE>MA</CAT_CODE>
    <NAME>Bad Guy 1</NAME>
    <ALTN_IDENTIFIC_TXT>BAD001</ALTN_IDENTIFIC_TXT>
    <OWNER_ID>7</OWNER_ID>
    <UPDATE_SEQNR>1</UPDATE_SEQNR>
  </OBJ_ITEM>
  <OBJ_ITEM>
    <OBJ_ITEM_ID>10</OBJ_ITEM_ID>
    <CAT_CODE>MA</CAT_CODE>
    <NAME>HMCS Grove</NAME>
    <ALTN_IDENTIFIC_TXT>CAD001</ALTN_IDENTIFIC_TXT>
    <OWNER_ID>7</OWNER_ID>
    <UPDATE_SEQNR>1</UPDATE_SEQNR>
  </OBJ_ITEM>
</OBJ_ITEM_TBL>
```

Figure 13. The object item content defines the objects to be used throughout the model. Here, two objects are described: item "28" which is the enemy materiel unit, and item "10" which is the HMCS Grove materiel unit.

The next section of the physical XML is shown in Figure 14. This section represents the materiel table of the LC2IEDB. At this stage, the relationships within the model are becoming evident. The materiel table cannot be loaded until after the object item table. This is because a foreign key relationship exists between the two tables linking the OBJ_ITEM_ID and the MAT_ID (relationships will be shown later).

The materiel table will be used in this example to store some of the submarine identification markings. Note that the record in Figure 14 shows the <MAT_ID> as "28" to match the <OBJ_ITEM_ID> in the OBJ_ITEM table.

The contact data (Table 7) initially identified the contact as "Orange with big yellow dot". In the LC2IEDB, this information has been split up among three tags. The <BODY_COLOUR_CODE> contains the code "ORANGE". The dot information is not longer explicit. The <MARKING_CODE> does not include a code for a dot. So, in this case a code "SYMBOL" is used. The <MARKING_COLOUR_CODE> indicates that the marking is "YELLOW".

```
<MAT_TBL>
  <MAT>
    <MAT_ID>28</MAT_ID>
    <SERIAL_NO_ID_TXT>Model 1234</SERIAL_NO_ID_TXT>
    <LOT_IDENTIFIC_TXT></LOT_IDENTIFIC_TXT>
    <BODY_COLOUR_CODE>ORANGE</BODY_COLOUR_CODE>
    <MARKING_CODE>SYMBOL</MARKING_CODE>
    <MARKING_COLOUR_CODE>YELLOW</MARKING_COLOUR_CODE>
    <OWNER_ID>7</OWNER_ID>
    <UPDATE_SEQNR>1</UPDATE_SEQNR>
  </MAT>
</MAT_TBL>
```


Figure 14. The materiel content that describes the markings on the enemy unit.

Next, the organisation table is loaded with two records. The organisation needs to be defined, because any reports placed in the LC2IEDB must identify the reporting organisation. Again, the relationships in the model require that the organisation table be loaded after the object item table. This is because a foreign key links one OBJ_ITEM record to many ORG records. The primary key of the ORG table is also the ORG_ID column.

Figure 15 shows the organisation records for both the object items. The first record containing <ORG_ID> "28" corresponds to <OBJ_ITEM_ID> "28" in OBJ_ITEM. This is the enemy submarine and as such is given the nickname "Bad Guy". In this table the category code represents the organisation class. Here, "UN" refers to a military unit.

The second record loaded into the table contains information regarding the Ownship. Again, the link back to the OBJ_ITEM table is via common values for the <ORG_ID> and the <OBJ_ITEM_ID>, that value being "10". In this record, the nickname is "CF" (representing Canadian Forces) and the organisation is again a military unit, as identified by "UN" in CAT_CODE.

```
<ORG_TBL>
  <ORG>
    <ORG_ID>28</ORG_ID>
    <CAT_CODE>UN</CAT_CODE>
    <NICKNAME_NAME>Bad Guy</NICKNAME_NAME>
    <OWNER_ID>7</OWNER_ID>
    <UPDATE_SEQNR>1</UPDATE_SEQNR>
  </ORG>
  <ORG>
    <ORG_ID>10</ORG_ID>
    <CAT_CODE>UN</CAT_CODE>
    <NICKNAME_NAME>CF</NICKNAME_NAME>
    <OWNER_ID>7</OWNER_ID>
    <UPDATE_SEQNR>1</UPDATE_SEQNR>
  </ORG>
</ORG_TBL>
```


Figure 15. The organisation content. Here, two records are used to define two organisations. The first organisation is <ORG_ID> "28" and is the enemy organisation. The second ID <ORG_ID> "10" is the ID of the Ownship (in this case the CF or Canadian Forces).

The next insertion in the LC2IEDB accounts for the type of objects. Object types form a central role in the LC2IEDM, but in our example they are not linked to any of the previous tables. The object type records identify two types of objects – a submarine and a vessel. The submarine is identified as <OBJ_TYPE_ID> "4", has nationality "BK" (representing Bosnia and Herzegovina). Note that the MIST system identified the submarine nationality as "BA" (see Figure 11). The change from "BA" to "BK" (see Figure 16) is due to the change in country code lists. The MIST system uses country codes from the ISO 3166 list, while the LC2IEDM uses the STANAG 1059 list. The object is described as a materiel type, as indicated by the <CAT_CODE> value of "MA".

The reporting platform type is described using the second record of the OBJ_TYPE table. The platform type is given the identifier of "3" and is described as a materiel type. The platform is noted to be a Canadian frigate, as described by the <NATIONAITY_CODE> of "CA" and <NAME> of "Frigate".

```
<OBJ_TYPE_TBL>
  <OBJ_TYPE>
    <OBJ_TYPE_ID>4</OBJ_TYPE_ID>
    <CAT_CODE>MA</CAT_CODE>
    <DUMMY_IND_CODE>YES</DUMMY_IND_CODE>
    <NAME>Sub</NAME>
    <NATIONALITY_CODE>BK</NATIONALITY_CODE>
    <OWNER_ID>7</OWNER_ID>
    <UPDATE_SEQNR>1</UPDATE_SEQNR>
  </OBJ_TYPE>
  <OBJ_TYPE>
    <OBJ_TYPE_ID>3</OBJ_TYPE_ID>
    <CAT_CODE>MA</CAT_CODE>
    <DUMMY_IND_CODE>YES</DUMMY_IND_CODE>
    <NAME>Frigate</NAME>
    <NATIONALITY_CODE>CA</NATIONALITY_CODE>
    <OWNER_ID>7</OWNER_ID>
    <UPDATE_SEQNR>1</UPDATE_SEQNR>
  </OBJ_TYPE>
</OBJ_TYPE_TBL>
```

Figure 16. The object type content is used to define objects. Here, two types are described: the enemy submarine as object "4" and the Ownship as object "3".

The next section is not presented in the same order as indicated in Annex 1. This is because it utilizes an enhancement to the XSLT that is used to convert from logical to physical XML tags. For software management purposes, all additions to the XSLT were placed at the end of the existing XSLT rather than being incorporated more logically in the XSLT structure. This results in the output physical XML being placed at the end of the XML document.

Figure 17 shows the addition of single records to each of three different tables. The purpose of this section is to identify that the Ownship has sonar capabilities. In the LC2IEDB, we must first establish that there is materiel associated with the object type. This is again a foreign key link between MAT_TYPE_ID and OBJ_TYPE_ID columns in the database tables. In the materiel type table, the fact that MAT_TYPE_ID "3" indicates that we are describing a characteristic of the OBJ_TYPE_ID "3". Recall that <OBJ_TYPE_ID> "3" is the Ownship (see Figure 16).

The materiel type table needs a record before inserting into the equipment type table. Similarly, the <EQPT_TYPE_ID> refers back to the <MAT_TYPE_ID>. The equipment is noted to be electronic, using the <CAT_CODE> of "ELCTRN".

Last, the electronic equipment type table identifies a relationship back to equipment type via the <ELCTRNC_EQPT_TYPE_ID> and the <EQPT_TYPE_ID>. The ELCTRNC_EQPT_TYPE content in Figure 17 represents the first content not in the same order as presented in Annex 1. The electronic equipment type is identified as a sensor using <CAT_CODE> of "SEN". The subcategory code identifies the electronic

equipment as “SONAR”. Note that the sonar frequency of 300 Hz (see Table 7) has no obvious place in the ELCTRNC_EQPT_TYPE table. The logical location for the 300 is unclear in the LC2IEDM.


```
<MAT_TYPE_TBL>
  <MAT_TYPE>
    <MAT_TYPE_ID>3</MAT_TYPE_ID>
    <CAT_CODE>NOS</CAT_CODE>
    <RPTBL_ITEM_TXT></RPTBL_ITEM_TXT>
    <STOCK_NO_TXT></STOCK_NO_TXT>
    <SUPPLY_CLASS_CODE></SUPPLY_CLASS_CODE>
    <OWNER_ID>7</OWNER_ID>
    <UPDATE_SEQNR>1</UPDATE_SEQNR>
  </MAT_TYPE>
</MAT_TYPE_TBL>
<EQPT_TYPE_TBL>
  <EQPT_TYPE>
    <EQPT_TYPE_ID>3</EQPT_TYPE_ID>
    <CAT_CODE>ELCTRN</CAT_CODE>
    <LOADED_WT_QTY></LOADED_WT_QTY>
    <UNLOADED_WT_QTY></UNLOADED_WT_QTY>
    <MAX_HEIGHT_DIM></MAX_HEIGHT_DIM>
    <MAX_LENGTH_DIM></MAX_LENGTH_DIM>
    <MAX_WIDTH_DIM></MAX_WIDTH_DIM>
    <OWNER_ID>7</OWNER_ID>
    <UPDATE_SEQNR>1</UPDATE_SEQNR>
  </EQPT_TYPE>
</EQPT_TYPE_TBL>
<ELCTRNC_EQPT_TYPE_TBL>
  <ELCTRNC_EQPT_TYPE>
    <ELCTRNC_EQPT_TYPE_ID>3</ELCTRNC_EQPT_TYPE_ID>
    <CAT_CODE>SEN</CAT_CODE>
    <SUBCAT_CODE>SONAR</SUBCAT_CODE>
    <OWNER_ID>7</OWNER_ID>
    <UPDATE_SEQNR>1</UPDATE_SEQNR>
  </ELCTRNC_EQPT_TYPE>
</ELCTRNC_EQPT_TYPE_TBL>
```


Figure 17. The material, equipment and electronic specifications for the sensor information.

The next section of the physical XML deals with the reporting table.   The reporting table has a central role in the LC2IEDB because this table stores timing information related to database inserts.

The information inserted into the RPTD table is shown in Figure 18. The reporting table primary key is the RPTD_ID. This must be unique and for this example the <RPTD_ID> is set to “8”. An <ACCURACY_CODE> indicates the expected accuracy of the information, with the “6” indicating that the “Basis for the estimate of

Reported data cannot be estimated" [19]. The <CAT_CODE> in this table provides information on the nature of the specific record. Here, "REP" indicates that the record "points to data based on fact or observation" [19]. The counting indicator code <CNTG_IND_CODE> is used to identify if the record is based on a count of objects. In this case, the record is not based on a count of objects, as indicated by "NO".

The credibility code indicates the "degree of trustworthiness of the data" record [19]. The code "RPTUNC" indicates the data "is open to or can be viewed with suspicion" [19]. The <RELIABILITY_CODE> indicates "the general appraisal of the source in graded terms to indicate the extent to which it has been proven it can be counted on or trusted in to do as expected" [19]. In this column the "A" indicates, "the source of the reported data can be considered as completely reliable i.e., erroneous information cannot be produced" [19].


```
<RPTD_TBL>
    <RPTD_ID>8</RPTD_ID>
    <ACCURACY_CODE>6</ACCURACY_CODE>
    <CAT_CODE>REP</CAT_CODE>
    <CNTG_IND_CODE>NO</CNTG_IND_CODE>
    <CREDIBILITY_CODE>RPTUNC</CREDIBILITY_CODE>
    <RELIABILITY_CODE>A</RELIABILITY_CODE>
    <REP_DATE>20031007</REP_DATE>
    <REP_TIME>111000</REP_TIME>
    <SOURCE_TYPE_CODE>CONTAC</SOURCE_TYPE_CODE>
    <TIMING_CAT_CODE>RDABST</TIMING_CAT_CODE>
    <REF_ID>18</REF_ID>
    <REP_ORG_ID>10</REP_ORG_ID>
    <ENT_CAT_CODE>OILOCA</ENT_CAT_CODE>
    <OWNER_ID>7</OWNER_ID>
    <UPDATE_SEQNR>1</UPDATE_SEQNR>
  </RPTD>
</RPTD_TBL>
```


Figure 18. The reporting data content as per the information in Table 7. Considerable new information has been added to this record to highlight the available detail of reporting.

The date and time content in the LC2IEDB are very different as compared to the MIST model. Here, the date and time is considered separately. As well, the date and time of the database insertion is considered separately from the date and time of the contact.

The <REP_DATE> field contains a date string in the form YYYYMMDD, where YYYY indicates the year, MM the month, and DD the day. The time is expressed as HHMMSS, where HH indicates the hour, MM the minute and SS the second. The RPTD table stores the date and time of the database record insertion, not the contact time.

It should be noted that the LC2IEDB allows any string placement in the REP_DATE (CHAR 8) and REP_TIME (CHAR 6) fields. This allows relative dates to be assigned but also reduces the models ability to automatically check date and time content.

Next in Figure 18 is the source type code. This code indicates the type of information source, or “the source type from which intelligence information is obtained and which is referred to by a specific” record [19]. In this example, “CONTAC” indicates a contact, and is formally defined as “the discrete airborne, surface or subsurface intelligence information is collected from electronic, acoustic, and/or visual sensors” [19]. The timing category code identifies if the date/time are absolute or relative. The “RDABST” indicates that the date/time are absolute.

The reference ID REF_ID is linked back to the REF_ID in the REF table (see Figure 12). Recall that the reference table provides the information source. The reporting organisation ID, REP_ORG_ID, provides a numeric identifier for the organisation. This organisation is responsible for providing the referenced data. The <REP_ORG_ID> of “10” refers back to the ORG information (see Figure 15) as the Canadian Forces.

The <REF_ID> tag in the RPTD table may be a NULL. In the NULL case, the zero aspect of the relationship to table REF is used, and in this case no record is required in REF. When <REF_ID> in RPTD has content, the one aspect of the relationship to the REF table forces the presence of a record in the REF table.

The ENT_CAT_CODE refers to the specific table associated with a particular reporting data record. This column allows an explicit reference to a table name, using a code. The exact functionality of this column is unclear. For this exercise, the column is filled with “OILOCA” indicating “Object Item Location”.

As note above, the RPTD table stores the date and time of the record insertion into the database. The record in Figure 18 indicated a record with absolute timing. In this case, the actual contact time is stored in the RPTD_ABS_TIMING table, as shown in Figure 19. In the LC2IEDB, the date and time are in the form described for the RPTD table, and are represented by values of 20030624 and 111000.

The RPTD_ABS_TIMING table also contains information on the duration and precision of the time values. The <DUR> tag provides the duration of time, in seconds, for which the report is valid. The <EFFCTV_TIME_PRECISION_CODE> provides an estimate of the precision to which the effective date and time are known.

```
  <RPTD_ABS_TIMING_TBL>
    <RPTD_ABS_TIMING>
      <RPTD_ABS_TIMING_RPTD_ID>8</RPTD_ABS_TIMING_RPTD_ID>
      <DUR>3600</DUR>
      <EFFCTV_DATE>20030623</EFFCTV_DATE>
      <EFFCTV_TIME>111000</EFFCTV_TIME>
<EFFCTV_TIME_PRECISION_CODE>SECOND</EFFCTV_TIME_PRECISION_CODE>
      <OWNER_ID>7</OWNER_ID>
      <UPDATE_SEQNR>1</UPDATE_SEQNR>
    </RPTD_ABS_TIMING>
  </RPTD_ABS_TIMING_TBL>
```


Figure 19. The reporting-data-absolute-timing table. The date-time of the contact information is stored in this table.

At this point it may be useful to pictorially examine the LC2IEDB tables used thus far in the investigation. These tables are presented in Figure 20. The figure shows the table names in uppercase characters above the table. A rectangular box illustrates a table. Column names and data types are indicted within the table. Primary keys are indicated as named columns above the horizontal separator line for the table and also by the key symbol. Below the separator line are non-key column names. Some of the non-key column names are designated as foreign keys, and are indicated by FK following the column name.

Relationships are illustrated using the Integration Definition for Information Modelling (IDEF1X) notation [23]. This notation illustrates a relationship between tables as either a solid or dashed line. The solid line represents a relationship where the foreign key is part of the primary key in the child table (an identifying relationship). A dashed line represents a relationship where the foreign key is not part of the primary key in the child table (a non-identifying relationship). A solid black circle on the end of a line indicates zero, one or more records. A diamond symbol on the end of a line indicates zero or one records. No symbol on the end of a line indicates one record. Finally, a red dot indicates a specialization. A specialization is a relationship that requires a record entry in a particular child table based on the content in the parent table (typically this content is in a category code column).

The figure illustrates an important point. The diagram and the table structure generated using the OCXS install do not exactly match. This is shown in the REF table, where the structure in Figure 12 contains <FORMAT_CODE> while the diagram does not. Such version control issues extend beyond this simple problem, to the extent of having multiple ERwin™ data models of the LC2IEDM with the same version and date, but different structure.

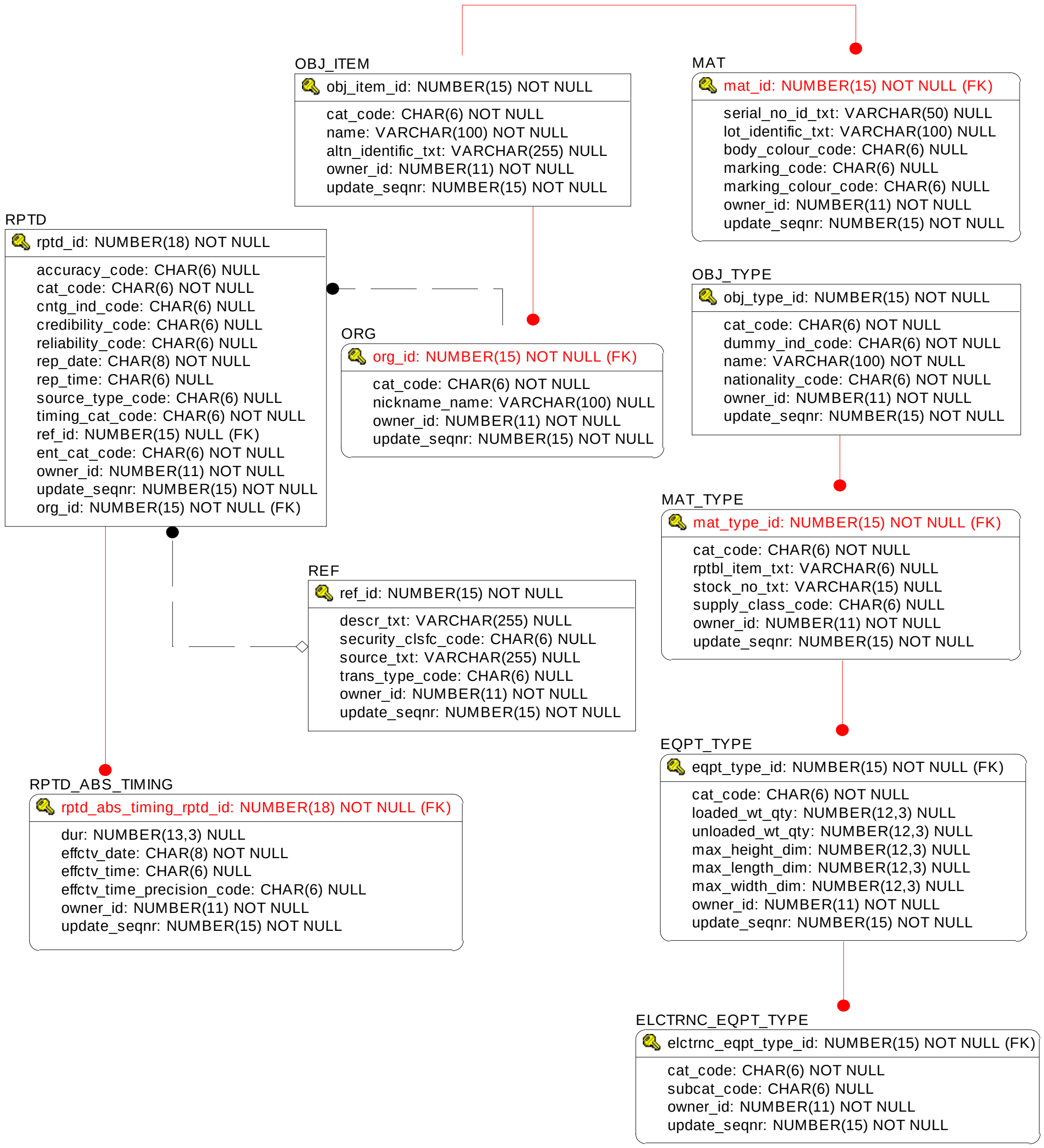

Figure 20. The table structure used thus far in the database inserts. The colours (in the electronic version) may be ignored. The rounded edge tables indicate dependent tables, while square edges indicate independent tables. Relationships are indicated using IDEF1X notation [23]. Key fields are indicated with the key symbol. Foreign keys are indicated using FK.

We proceed by examining the XML that describes the status of the object (Figure 21). In this case, the status of the enemy submarine is described using the OBJ_ITEM_STAT table. The <OBJ_ITEM_ID> column is related back to the object item table (see Figure 13) where the item ID "28" is used to define the enemy object.

The object item status index <OBJ_ITEM_STAT_IX> represents a unique character string, which when combined with the object item ID, forms the primary key of the table. This means that the combined ID and index allows for the possibility of many different status reports for a single object item.

The category code for the OBJ_ITEM_STAT table represents a class of object. The record in Figure 21 indicates "MA", which identifies a Materiel. The hostility code <HSTLT_CODE> indicates "the perceived status of a specific OBJECT-ITEM" [19] and in this case is set to "HO" representing "positively identified as an enemy" [19]. The booby trap indicator code <BOOBY_TRAP_IND_CODE> describes if the "specific object-item has been booby-trapped" [19]. The code of "NO" indicates that it has not been booby-trapped.

The <RPTD_ID> provides a link back to the RPTD table (see Figure 18). This provides a link between the objects status and the report made on the object.


```
<OBJ_ITEM_STAT_TBL>
  <OBJ_ITEM_STAT>
    <OBJ_ITEM_ID>28</OBJ_ITEM_ID>
    <OBJ_ITEM_STAT_IX>277</OBJ_ITEM_STAT_IX>
    <CAT_CODE>MA</CAT_CODE>
    <HSTLY_CODE>HO</HSTLY_CODE>
    <BOOBY_TRAP_IND_CODE>NO</BOOBY_TRAP_IND_CODE>
    <RPTD_ID>8</RPTD_ID>
    <OWNER_ID>7</OWNER_ID>
    <UPDATE_SEQNR>1</UPDATE_SEQNR>
  </OBJ_ITEM_STAT>
</OBJ_ITEM_STAT_TBL>
```


Figure 21. The Object Item Status table describes the perceived status of the object. In this example, the status of the enemy contact is described.

The location table (Figure 22) provides the location categorisation. Note that the table does not provide location information pertaining to a particular object, but rather an index for specific location information. The table defines a LOC_ID to identify a particular location, and a category code to describe the type of location. Figure 22 provides information on point, surface and volume locations, denoted by "PT", "SURFAC" and "VL" in the <CAT_CODE>.

The point in the LOC table denoted <LOC_ID> 5 will be used to describe a location for Ownship. The point is therefore described using the POINT and ABS_POINT tables in Figure 23. The POINT_TBL provides a <POINT_ID> that links back to the <LOC_ID> in the LOC_TBL. The POINT <CAT_CODE> identifies the category of the point. In this case, an absolute point denoted "ABS".

The ABS_POINT table is used to describe the absolute location of the point. In this case, the location of <ABS_POINT_ID> "5" is described. This location has a latitude coordinate of "45" and a longitude coordinate of "-50". These columns are defined as positive northward and positive eastward. The angular precision code provides the resolution of the position in degrees and minutes. Here, we declare a precision of one minute.

```
<LOC_TBL>
  <LOC>
    <LOC_ID>5</LOC_ID>
    <CAT_CODE>PT</CAT_CODE>
    <OWNER_ID>7</OWNER_ID>
    <UPDATE_SEQNR>1</UPDATE_SEQNR>
  </LOC>
  <LOC>
    <LOC_ID>83</LOC_ID>
    <CAT_CODE>VL</CAT_CODE>
    <OWNER_ID>7</OWNER_ID>
    <UPDATE_SEQNR>1</UPDATE_SEQNR>
  </LOC>
  <LOC>
    <LOC_ID>6</LOC_ID>
    <CAT_CODE>SURFAC</CAT_CODE>
    <OWNER_ID>7</OWNER_ID>
    <UPDATE_SEQNR>1</UPDATE_SEQNR>
  </LOC>
</LOC_TBL>
```

Figure 22. The Location content defines high-level information on locations. In this example, point, surface and volume locations are described.

<LOC_ID> values 6 and 83 will be used to define the location of the contact. <LOC_ID> 6 defines a surface as denoted by <CAT_CODE> "SURFAC". The SURFACE table (Figure 24) defines a <SURFACE_ID> of "6" which is related back to the <LOC_ID> in the LOC table. The category code in surface is described as "FA" meaning a fan or sector shaped area.

The <FAN_AREA_VERTEX_POINT_ID> provides a relation back to the POINT_TBL, in <POINT_ID>. This then describes the vertex for the fan area. In this case, <POINT_ID> "5" is a particular position as defined in Figure 23.

```
<POINT_TBL>
  <POINT>
    <POINT_ID>5</POINT_ID>
    <CAT_CODE>ABS</CAT_CODE>
    <OWNER_ID>7</OWNER_ID>
    <UPDATE_SEQNR>1</UPDATE_SEQNR>
  </POINT>
</POINT_TBL>
<ABS_POINT_TBL>
  <ABS_POINT>
    <ABS_POINT_ID>5</ABS_POINT_ID>
    <LAT_COORD>45</LAT_COORD>
    <LONG_COORD>-50</LONG_COORD>
    <ANGULAR_PRECISION_CODE>MINUTE</ANGULAR_PRECISION_CODE>
    <ABS_POINT_VER_DIST_ID></ABS_POINT_VER_DIST_ID>
    <OWNER_ID>7</OWNER_ID>
    <UPDATE_SEQNR>1</UPDATE_SEQNR>
  </ABS_POINT>
</ABS_POINT_TBL>
```


Figure 23. The PT record in the LOC table describes a point. The tables POINT and ABS_POINT further describe the point.

The FAN_AREA (Figure 24) table is used to describe the fan shape. The <FAN_AREA_ID> is also given a value of "6" and is related to the <SURFACE_ID> in the SURFACE table. The minimum and maximum range dimensions (<MNM_RANGE_DIM> and <MAX_RANGE_DIM>) are based on the contact information (Table 7). The information describes the contact at a range of 12 km with a range error of ± 0.5 km. As well, the FAN_AREA record describes the angular extent of the fan, that being 263.25° to 276.75° T. This corresponds to the Table 7 description of 270°T ± 6.75°. The angular description in Figure 24 uses <ORIENT_ANGLE> to define the angle to the first edge of the fan area. The fan width, <SECTOR_SIZE_ANGLE>, is then 13.5°.

```
<SURFACE_TBL>
  <SURFACE>
    <SURFACE_ID>6</SURFACE_ID>
    <CAT_CODE>FA</CAT_CODE>
    <OWNER_ID>7</OWNER_ID>
    <UPDATE_SEQNR>1</UPDATE_SEQNR>
  </SURFACE>
</SURFACE_TBL>
<FAN_AREA_TBL>
  <FAN_AREA>
    <FAN_AREA_ID>6</FAN_AREA_ID>
    <MNM_RANGE_DIM>11.5</MNM_RANGE_DIM>
    <MAX_RANGE_DIM>12.5</MAX_RANGE_DIM>
    <ORIENT_ANGLE>263.25</ORIENT_ANGLE>
    <SECTOR_SIZE_ANGLE>13.5</SECTOR_SIZE_ANGLE>
    <FAN_AREA_VERTEX_POINT_ID>5</FAN_AREA_VERTEX_POINT_ID>
    <OWNER_ID>7</OWNER_ID>
    <UPDATE_SEQNR>1</UPDATE_SEQNR>
  </FAN_AREA>
</FAN_AREA_TBL>
```

Figure 24. The location identified using LOC_ID “6” is described further using tables SURFACE and FAN_AREA.

Now consider <LOC_ID> 83, which is defined as a volume (Figure 22). Due to relationship requirements, we need to define the vertical extent of the volume before actually defining the volume. We do this by using the VER_DIST table as shown in Figure 25. Here, the VER_DIST_ID points 81 and 82 are defined with the vertical values of “110” and “90” m. The vertical values are given in <DIM> in the VER_DIST table. These values correspond to the depth of the contact (see Table 7, 100 m ± 10m).

It should be noted that these vertical values are positive, and therefore properly refer to values above sea level. The LC2IEDB Version 5.3 does not permit negative values. This has been changed in Version 6 of the model, where negative values are permitted [24].

```
<VER_DIST_TBL>
  <VER_DIST>
    <VER_DIST_ID>81</VER_DIST_ID>
    <CAT_CODE>HEIGHT</CAT_CODE>
    <DIM>110</DIM>
    <PRECISION_CODE>1M</PRECISION_CODE>
    <OWNER_ID>7</OWNER_ID>
    <UPDATE_SEQNR>1</UPDATE_SEQNR>
  </VER_DIST>
  <VER_DIST>
    <VER_DIST_ID>82</VER_DIST_ID>
    <CAT_CODE>HEIGHT</CAT_CODE>
    <DIM>90</DIM>
    <PRECISION_CODE>1M</PRECISION_CODE>
    <OWNER_ID>7</OWNER_ID>
    <UPDATE_SEQNR>1</UPDATE_SEQNR>
  </VER_DIST>
</VER_DIST_TBL>
```

Figure 25. The vertical values of 110 and 90 m are described in the VER_DIST table. These values are positive because the LC2IEDM Version 5.3 model did not accept negative values for the vertical distance DIM field. This has been corrected in Version 6 [24].

Next, we construct a geometric volume (Figure 26) that is based on the surface fan area and the vertical distance points. The GEOM_VOL_ID is linked back to the LOC_ID. As well, the UPPER and LOWER_VER_DIST_ID are linked to the VER_DIST_IDs in the VER_DIST table (Figure 25). The SURFACE_VOL table provides a link between SURFACE_VOL_ID and the GEOM_VOL_ID. The SURFACE_VOL record also describes a defining surface, which defines the surface of the volume. The <SURFACE_VOL_DFNG_SURFACE_ID> provides a link to a particular surface, in this case the surface with <SURFACE_ID> 6 in the SURFACE table.

At this point, we should summarize what we have created. The model has allowed the definition of a geometric volume using a surface and a depth interval. Here, we have used it to define a fan-area slice of the ocean, with a range of 11.5 to 12.5 km, a bearing of 263.25 to 276.75 and a depth of 90 to 110 m. The fan area is relative to a known vertex point, that being position 45°N, -50°W

```
  <GEOM_VOL_TBL>
    <GEOM_VOL>
      <GEOM_VOL_ID>83</GEOM_VOL_ID>
      <CAT_CODE>SURVOL</CAT_CODE>
<GEOM_VOL_LOWER_VER_DIST_ID>82</GEOM_VOL_LOWER_VER_DIST_ID>
<GEOM_VOL_UPPER_VER_DIST_ID>81</GEOM_VOL_UPPER_VER_DIST_ID>
      <OWNER_ID>7</OWNER_ID>
      <UPDATE_SEQNR>1</UPDATE_SEQNR>
    </GEOM_VOL>
  </GEOM_VOL_TBL>
  <SURFACE_VOL_TBL>
    <SURFACE_VOL>
      <SURFACE_VOL_ID>83</SURFACE_VOL_ID>
      <OWNER_ID>7</OWNER_ID>
      <UPDATE_SEQNR>1</UPDATE_SEQNR>
<SURFACE_VOL_DFNG_SURFACE_ID>6</SURFACE_VOL_DFNG_SURFACE_ID>
    </SURFACE_VOL>
  </SURFACE_VOL_TBL>
```

Figure 26. The geometric volume describes a surface volume. Surface volumes are based on a previous description of a surface.

At this stage the locations have been defined, but have not yet been linked to the objects. For the linking, we use the object item location table (OBJ_ITEM_LOC) as shown in Figure 27. The first record in Figure 27 links <LOC_ID> "83" to object item "28". This is linking the enemy contact (Figure 13) with the location described as a geometric volume (Figure 22).

The primary key for this table is a composite, made up of LOC_ID, OBJ_ITEM_ID and the object item location index OBJ_ITEM_LOC_IX. The index is present to allow an object to revisit a single location. The accuracy quantity is defined as "the non-monetary numeric value representing the uncertainty in the estimate of a specific OBJECT-ITEM-LOCATION, expressed in units of metres". It is unclear how a monetary value could represent an uncertainty in an object item location. This definition illustrates a need for clearer definitions with the LC2IEDM.

For the object denoted as <OBJ_ITEM_ID> "28", we use the <BEARING_ANGLE> and <SPEED_RATE> data directly from the contact data. This is because item "28" is the contact for the enemy submarine. Here, <BEARING_ANGLE> is defined as "the rotational measurement clockwise from the line of true North to the direction of motion of a specific OBJECT-ITEM at a specific LOCATION". This is clearly the direction of object motion (a direction of 90° based on the data in Table 7). <SPEED_RATE> is defined in km/hr and is therefore given the value 18 (see Table 7).

```
<OBJ_ITEM_LOC_TBL>
  <OBJ_ITEM_LOC>
    <LOC_ID>83</LOC_ID>
    <OBJ_ITEM_ID>28</OBJ_ITEM_ID>
    <OBJ_ITEM_LOC_IX>1</OBJ_ITEM_LOC_IX>
    <ACCURACY_QTY>10</ACCURACY_QTY>
    <BEARING_ANGLE>90</BEARING_ANGLE>
    <SPEED_RATE>18</SPEED_RATE>
    <GEOMETRY_CFEAT_TYPE_ID></GEOMETRY_CFEAT_TYPE_ID>
    <RPTD_ID>8</RPTD_ID>
    <OWNER_ID>7</OWNER_ID>
    <UPDATE_SEQNR>1</UPDATE_SEQNR>
  </OBJ_ITEM_LOC>
  <OBJ_ITEM_LOC>
    <LOC_ID>5</LOC_ID>
    <OBJ_ITEM_ID>10</OBJ_ITEM_ID>
    <OBJ_ITEM_LOC_IX>13</OBJ_ITEM_LOC_IX>
    <ACCURACY_QTY>10</ACCURACY_QTY>
    <BEARING_ANGLE></BEARING_ANGLE>
    <SPEED_RATE></SPEED_RATE>
    <GEOMETRY_CFEAT_TYPE_ID></GEOMETRY_CFEAT_TYPE_ID>
    <RPTD_ID>8</RPTD_ID>
    <OWNER_ID>7</OWNER_ID>
    <UPDATE_SEQNR>1</UPDATE_SEQNR>
  </OBJ_ITEM_LOC>
</OBJ_ITEM_LOC_TBL>
```


Figure 27. The object item location content is used to link the object and the location information.

The pictorial view of the location-related tables is shown in Figure 28. The figure shows the intricate assortment of relationships that must be obeyed during data insertions. Although sometimes difficult to follow, the relationships provide considerable flexibility.

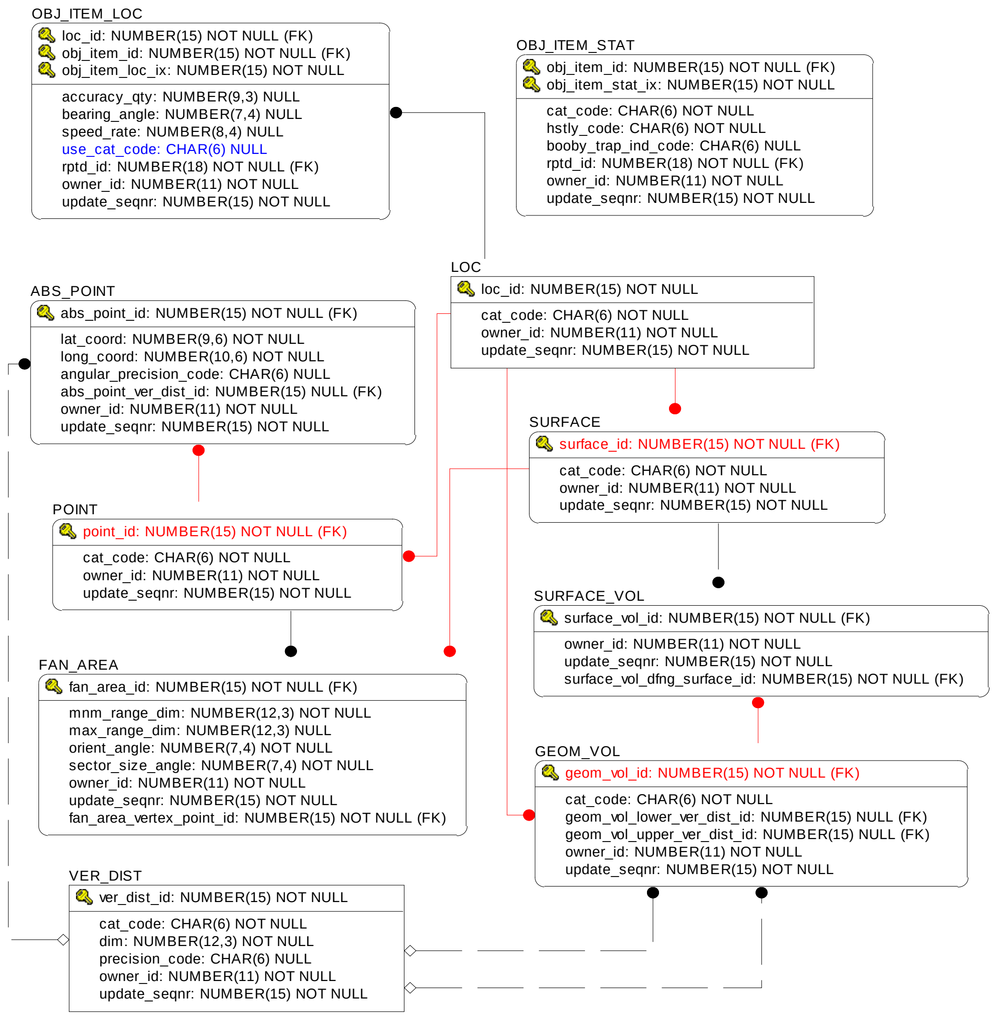

Figure 28. The locations identified using LOC_ID are further described using tables SURFACE, POINT and GEOM_VOL.

Figure 29 is the next XML snippet that will add the fact that the enemy is a subsurface unit. The capability table (CAPAB) identifies a particular capability using <CAPAB_ID> “77”. The category code in CAPAB indicates “MOBL” which is loosely defined as “the nominal ability to move in air, on water, under water or over a specific type of terrain” [19]. The <DAY_NIGHT_CODE> indicates that the unit is capable of both day and night operations. The <UOM_CODE> “represents the quantities in terms of which the magnitude of a CAPABILITY category is stated” [19]. In this case, <UOM_CODE> is “EA” representing each.

```
<CAPAB_TBL>
  <CAPAB>
    <CAPAB_ID>77</CAPAB_ID>
    <CAT_CODE>MOBL</CAT_CODE>
    <SUBCAT_CODE></SUBCAT_CODE>
    <DAY_NIGHT_CODE>DN</DAY_NIGHT_CODE>
    <UOM_CODE>EA</UOM_CODE>
    <OWNER_ID>7</OWNER_ID>
    <UPDATE_SEQNR>1</UPDATE_SEQNR>
  </CAPAB>
</CAPAB_TBL>
<MOB_CAPAB_TBL>
  <MOB_CAPAB>
    <MOB_CAPAB_ID>77</MOB_CAPAB_ID>
    <CAT_CODE>SEASS</CAT_CODE>
    <TERRAIN_TYPE_CODE>NOS</TERRAIN_TYPE_CODE>
    <OWNER_ID>7</OWNER_ID>
    <UPDATE_SEQNR>1</UPDATE_SEQNR>
  </MOB_CAPAB>
</MOB_CAPAB_TBL>
<OBJ_ITEM_CAPAB_TBL>
  <OBJ_ITEM_CAPAB>
    <OBJ_ITEM_ID>28</OBJ_ITEM_ID>
    <CAPAB_ID>77</CAPAB_ID>
    <OBJ_ITEM_CAPAB_IX>23</OBJ_ITEM_CAPAB_IX>
    <MSN_PRIMACY_CODE>PRIME</MSN_PRIMACY_CODE>
    <QTY>1</QTY>
    <RPTD_ID>8</RPTD_ID>
    <OWNER_ID>7</OWNER_ID>
    <UPDATE_SEQNR>1</UPDATE_SEQNR>
  </OBJ_ITEM_CAPAB>
</OBJ_ITEM_CAPAB_TBL>
```

Figure 29. The capability and mobility of the enemy unit.

The mobility of the capability is then described using the MOB_CAPAB table. The <MOB_CAPAB_ID> provides a relation back to the <CAPAB_ID> in the CAPAB table. The <CAT_CODE> in MOB_CAPAB indicates “the type of mobility by land, air or sea to which a specific MOBILITY-CAPABILITY pertains” [19]. In this case “SEASS” indicates “the capability of a device to move on or under the sea surface” [19]. The terrain type code is used to describe the class of terrain that pertains to this mobility. There is no current value that seems appropriate for a submarine, and so this code is set to “NOS” indicating, “the appropriate value is not in the set of specified values” [19].

The object item capability table provides the link between the capability and the object item. The <OBJ_ITEM_ID> “28” is related back to the object item table (Figure 13) and the <CAPAB_ID> “77” is related to the CAPAB table. The object item capability

index provides a simple numeric index for the record in the OBJ_ITEM_CAPAB table, thereby allowing the evolution of the capability for the object. The mission primacy code identifies the level of capability as prime, second or third. The <QTY> or quantity field indicates the number of units while the <RPTD_ID> provides a relation back to the reporting data table.

To deal with the composite referred to in the contact information (Table 7), we must implement a fusion of information within the LC2IEDM. The fusion is implemented in the terms of a context – essentially putting the two original records in the context of a third.

In this investigation, the original information is noted to be a contact that has been based on a composite of contact number 1 and contact number 4. In the LC2IEDB, the original contact data has been denoted with the <RPTD_ID> of "8". To represent this RPTD_ID as a composite of two other records, in this case records 1 and 4, we must first create report record 1 and 4 in the RPTD table. These two records are shown in Figure 30.

The two records in Figure 30 are given only as a representation of the required records. The records are complete only to the point necessary to meet the integrity constraints of the database. Note that both records essentially contain identical information.

After the reporting data has been added to the RPTD table, the actual context can be added. This is shown in Figure 31. The CONTXT table describes the context, in this case denoted by <CONTXT_ID> "1" with the <NAME> of "Composite of Tracks 1 and 4". Next, the various elements of the context are described. This particular context is made up of two elements. Element index 1 of the context, <CONTXT_ELMT_IX>, refers to the reporting data record "1", while element index "2" refers to the reporting data record "4". Finally, creating an association between the context and the reporting data records completes the fusion. Using the <CONTXT_ID>, the <RPTD_ID> and <CAT_CODE> in the CONTXT_RPTD_ASSOC table, we may state that <CONTXT_ID> "1" implies (CAT_CODE of "IMPL") the <RPTD_ID> "8".

```
<RPTD_TBL>
  <RPTD>
    <RPTD_ID>1</RPTD_ID>
    <ACCURACY_CODE></ACCURACY_CODE>
    <CAT_CODE>REP</CAT_CODE>
    <CNTG_IND_CODE></CNTG_IND_CODE>
    <CREDIBILITY_CODE></CREDIBILITY_CODE>
    <RELIABILITY_CODE></RELIABILITY_CODE>
    <REP_DATE>20031007</REP_DATE>
    <REP_TIME></REP_TIME>
    <SOURCE_TYPE_CODE></SOURCE_TYPE_CODE>
    <TIMING_CAT_CODE>RDABST</TIMING_CAT_CODE>
    <REF_ID>18</REF_ID>
    <REP_ORG_ID>10</REP_ORG_ID>
    <ENT_CAT_CODE>OILOCA</ENT_CAT_CODE>
    <OWNER_ID>7</OWNER_ID>
    <UPDATE_SEQNR>1</UPDATE_SEQNR>
  </RPTD>
  <RPTD>
    <RPTD_ID>4</RPTD_ID>
    <ACCURACY_CODE></ACCURACY_CODE>
    <CAT_CODE>REP</CAT_CODE>
    <CNTG_IND_CODE></CNTG_IND_CODE>
    <CREDIBILITY_CODE></CREDIBILITY_CODE>
    <RELIABILITY_CODE></RELIABILITY_CODE>
    <REP_DATE>20031007</REP_DATE>
    <REP_TIME></REP_TIME>
    <SOURCE_TYPE_CODE></SOURCE_TYPE_CODE>
    <TIMING_CAT_CODE>RDABST</TIMING_CAT_CODE>
    <REF_ID>18</REF_ID>
    <REP_ORG_ID>10</REP_ORG_ID>
    <ENT_CAT_CODE>OILOCA</ENT_CAT_CODE>
    <OWNER_ID>7</OWNER_ID>
    <UPDATE_SEQNR>1</UPDATE_SEQNR>
  </RPTD>
</RPTD_TBL>
```


Figure 30. The reporting data content required to support the composite information.

```
  &lt;CONTXT_TBL&gt;
    &lt;CONTXT&gt;
      &lt;CONTXT_ID&gt;1&lt;/CONTXT_ID&gt;
      &lt;NAME&gt;Composite of Tracks 1 and 4&lt;/NAME&gt;
      &lt;OWNER_ID&gt;7&lt;/OWNER_ID&gt;
      &lt;UPDATE_SEQNR&gt;1&lt;/UPDATE_SEQNR&gt;
    &lt;/CONTXT&gt;
  &lt;/CONTXT_TBL&gt;
  &lt;CONTXT_RPTD_ASSOC_TBL&gt;
    &lt;CONTXT_RPTD_ASSOC&gt;
      &lt;CONTXT_ID&gt;1&lt;/CONTXT_ID&gt;
      &lt;RPTD_ID&gt;8&lt;/RPTD_ID&gt;
      &lt;CONTXT_RPTD_ASSOC_IX&gt;1&lt;/CONTXT_RPTD_ASSOC_IX&gt;
      &lt;CAT_CODE&gt;IMPL&lt;/CAT_CODE&gt;
      &lt;OWNER_ID&gt;7&lt;/OWNER_ID&gt;
      &lt;UPDATE_SEQNR&gt;1&lt;/UPDATE_SEQNR&gt;
    &lt;/CONTXT_RPTD_ASSOC&gt;
  &lt;/CONTXT_RPTD_ASSOC_TBL&gt;
  &lt;CONTXT_ELMT_TBL&gt;
    &lt;CONTXT_ELMT&gt;
      &lt;CONTXT_ID&gt;1&lt;/CONTXT_ID&gt;
      &lt;CONTXT_ELMT_IX&gt;1&lt;/CONTXT_ELMT_IX&gt;
      &lt;RPTD_ID&gt;1&lt;/RPTD_ID&gt;
      &lt;OWNER_ID&gt;7&lt;/OWNER_ID&gt;
      &lt;UPDATE_SEQNR&gt;1&lt;/UPDATE_SEQNR&gt;
    &lt;/CONTXT_ELMT&gt;
    &lt;CONTXT_ELMT&gt;
      &lt;CONTXT_ID&gt;1&lt;/CONTXT_ID&gt;
      &lt;CONTXT_ELMT_IX&gt;2&lt;/CONTXT_ELMT_IX&gt;
      &lt;RPTD_ID&gt;4&lt;/RPTD_ID&gt;
      &lt;OWNER_ID&gt;7&lt;/OWNER_ID&gt;
      &lt;UPDATE_SEQNR&gt;1&lt;/UPDATE_SEQNR&gt;
    &lt;/CONTXT_ELMT&gt;
  &lt;/CONTXT_ELMT_TBL&gt;
&lt;/GH5Complete&gt;
```

Figure 31. The composite information. Context elements 1 and 2 refer to reported data records 1 and 4. These context elements imply reported data record 8.

The table structures used for the capabilities and establishing the context are shown in Figure 32. Of interest in the fusion aspect of the structure, are the tables related to the context. Figure 32 shows that it is not necessary to insert records into CONTXT_ELMT as indicated by the relationships. It is unclear what the CONTXT_ELMT records contribute to the information content of the database as the information in these records is contained in the CONTXT_RPTD_ASSOC table.

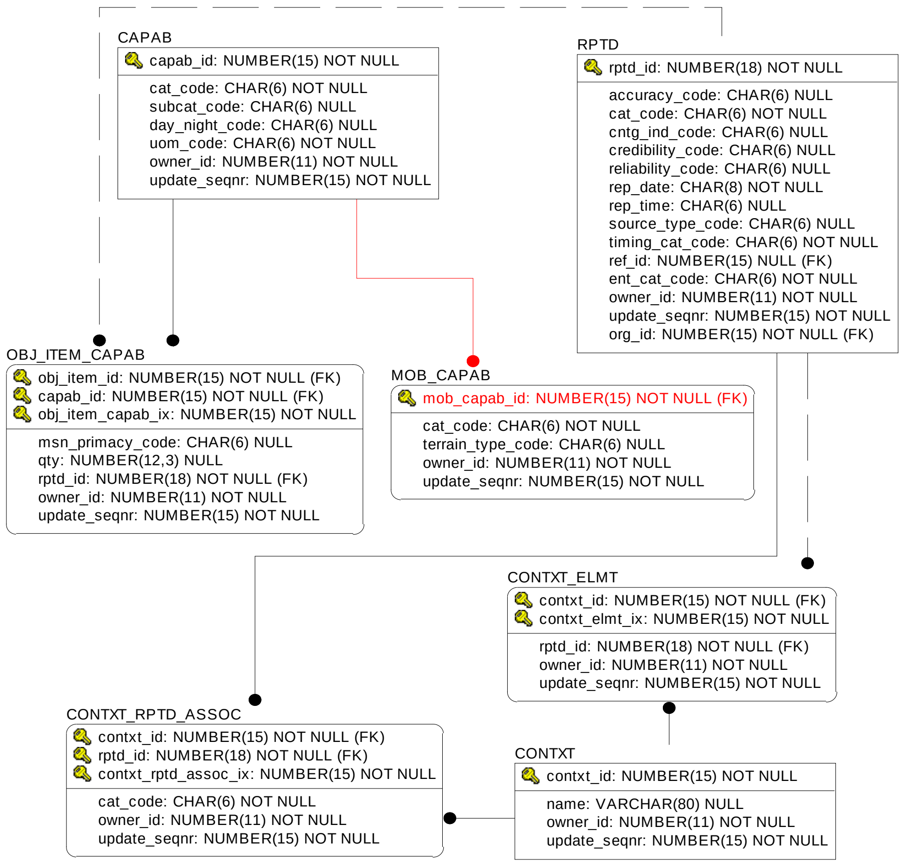

Figure 32. The data fusion is accomplished within the LC2IEDB using the CONTXT table. Individual reports in RPTD are related to records in CONTXT using an association established in CONTXT_RPTD_ASSOC.

7.3 Issues Related to Information Placement into the LC2IEDB

There are four components of the contact information that have not been placed in the LC2IEDB. The lack of placement can be attributed to one of two reasons: i) the contact information does not appear to have a logical placement location in the LC2IEDB, or ii) the contact information belongs in another component of the larger LC2IEDB system. This section summarises the information that has not been placed in the LC2IEDB.

7.3.1 SonarFrequencies

The specific frequency information of the operating sonar (300 Hz in Table 7) does not appear to have a logical storage location in the LC2IEDM. The sonar equipment information has been placed in the ELCTRNC_EQPT_TYPE table (see Figure 17) but this particular table has no available columns for information specific to the equipment. To store this type of information, an extension to the data model is required.

7.3.2 SecurityClassification and UnRestrictedReleaseList

Security issues have not been incorporated directly into the LC2IEDM. However, the communications layer between instances of LC2IEDBs does deal with security. This layer, called the Automated Replication Mechanism (ARM) [25] consists of software and a database that controls the flow of information between LC2IEDB instances. The ARM is currently being investigated.

7.3.3 Altitude

The vertical position of the submarine contact presents a problem. As noted previously, the vertical distances contained in VER_DIST are all positive because the database does not allow negative values. This problem was ignored during this exercise because the LC2IEDM Version 6 allows negative vertical distances [24].

7.3.4 Other Notable Issues

The following issues were raised in the main body of this report, and are summarized here for convenience.

1. OCXS and LC2IEDM Version 5.3 do not exactly match. This is shown in the REF table, where the structure in Figure 12 contains <FORMAT_CODE> while the Version 5.3 data model does not.

2. Several definitions within the LC2IEDM describe “non-monetary units”. For example, the ACCURACY_QTY of OBJ_ITEM_LOC is described as “the non-monetary numeric value representing the uncertainty in the estimate of a specific OBJECT-ITEM-LOCATION, expressed in units of metres”. Describing a value of metres as non-monetary is somewhat confusing. It is natural for readers to associate monetary with money or exchange currency. The definitions within the LC2IEDM need to be reviewed for this type of confusion.

3. The RPTD table column ENT_CAT_CODE requires more thorough explanation. The present documentation does describe the columns content, but does not clearly describe the columns intent.

8. Concluding Remarks

This investigation has concentrated on understanding both the MIST and LC2IEDM data structures as it relates to sonar contact data. The investigation has described the software required by the systems and how the user can setup and test the software implementation on their particular computers.

The focus of this effort was on placing sonar contact data into the MIST and LC2IEDB systems. The placement of such a concrete data unit provided an opportunity to examine in detail both the structure and documentation associated with both systems.

Regarding the MIST system, the important points that have resulted from this investigation are centred on the structure and documentation of the contact record. The MIST contact record structure is in need of re-examination using rigorous definitions. The structure could be defined using a variety of techniques, such as XML schema or a data model. In this exercise, the structure needs to be re-examined to address various internal inconsistencies. These inconsistencies are a result of data unit existence being dependent on the occurrence of other data units.

The MIST record re-examination should also take into account definitions for the data units. The developers need to clarify many tag definitions in the MIST documentation. For example, the definitions for the error tags <RangeError>, <BearingError> and <PositionError> are currently problematic.

The LC2IEDM also has issues that need to be resolved. For example, as an Army-developed system it is lacking an underwater perspective. This is evident in tables related to vertical positioning, and mobility. The datums associated with the vertical positioning also need to be examined from the underwater perspective.

Other underwater issues cannot be clearly resolved using the existing structure. For example, an underwater profile (e.g., a sound velocity profile) does not currently have a logical home in the data model. However, such data could be placed in the model by utilizing the context assessment table, CONTXT_ASSESS. By first defining the context, we could place the profile information into the TXT column of the CONTXT_ASSESS table. However, this would require agreement among the parties involved on the structure and semantics of the passed information. For example, the agreed encoding could declare the first data item as the number of depth sound speed pairs, followed by a series of comma separated pairs.

Such a solution is really an example of forcing data units into a data model, when the model was not designed to store the data units. A better solution is a model extension, where tables are added to the existing structure to specifically address maritime requirements. However, this option also has implications for the replication component of the system. The possibility of database extensions is currently being explored.

The OCXS component of the LC2IEDB system also requires effort in a variety of directions. The OCXS is in need of formal documentation that explains the various components of the system. The present level of documentation does not describe the system in sufficient detail for the novice user.

The error reporting from the OCXS system also should be improved. At present, errors in the XML input are very difficult to trace and the reporting errors from the OCXS does not indicate the table or column where the error occurred. Rather, OCXS reports the SQL error that results. Often, the SQL error indicates violation of a particular integrity constraint which is of minimal use when attempting to input data involving many tables.

9. References

1. Stokes, Donald E. 1997. Pasteur's Quadrant, Basic Science and Technological Innovation, Brookings Institute Press, Washington DC, 180pp.

2. Longard, John R., 1993. Knots, Volts and Decibels, An Informal History of the Naval Research Establishment, 1940-1967, Defence Research Establishment Atlantic.

3. Leadmark, the Navy's Strategy for 2020 – A Summary. Directorate of Maritime Strategy, NDHQ/Chief of the Maritime Staff, ISBN 0-662-65977-5.

4. The Land C2 Information Exchange Data Model, (ATCCIS) Baseline 2.0+, 18 February 2003, Data and Procedures Working Group, Greding, Germany. (Interim Draft).

5. See **www.oracle.com**

6. See **http://www.oracle.com/corporate/annual_report/2001/index.html?intro.html**

7. Vaas, Lisa. 2003. Oracle's Database Lead Slips, eWeek, see **http://www.eweek.com/article2/0,3959,923436,00.asp**

8. SQL*Plus Users Guide and Reference, Release 9.0.1, July 2001.

9. See **http://www.postgresql.org/**

10. PostgreSQL 7.2 User's Guide, by The PostgreSQL Global Development Group, Copyright © 1996-2001 by The PostgreSQL Global Development Group.

11. See **http://www.pgaccess.org/**

12. See **http://www.apache.org/**

13. See **http://www.w3.org/2000/xp/Group/**

14. Isenor, Anthony W. 2003. XML Based Manipulation of Codes Exchanged Between Data Systems, DRDC Atlantic TM 2003-132, Defence R&D Canada – Atlantic.

15. See **http://www.w3.org/TR/SOAP/**

16. Englander, Robert. 2002. Java and SOAP, O'Reilly and Associates, Inc., 1005 Gravenstein Highway North, Sebastopol, CA 95272, USA.

17. See **http://ca.com/**

18. Burkley, Frederick G. Integrated Computing Environment, Operational Context Exchange Service, October 30, 2002.

19. The Generic Hub 5 Data Model – Annexes, Baseline 1.0 (Draft 0.9), 30 July 2002.

20. Coalition Data Server Installation Guide, November 20, 2002.

21. CDS Coalition Data Server, Programmers Reference Guide, September 26, 2002, Version 1.0, Naval Undersea Warfare Center Division Newport.

22. **http://www.iso.org/iso/en/prods-services/iso3166ma/index.html**

23. see **http://www.essentialstrategies.com/publications/modeling/idef1x.htm**

24. LCol. Jacques Hamel, Personal Communication, Oct. 1, 2003.

25. Automated Replication Mechanism (ARM), SADI System Interface Control Document Volume 4, Baseline 1.0 (Draft 0.9), 30 July 2002.

Annex 1

The following is the contiguous XML document used to load the LC2IEDB with the contact record content.


```
<?xml version='1.0' ?>
<GH5Complete xmlns:dt="urn:schemas-microsoft-com:datatypes">
  <REF_TBL>
    <REF>
      <REF_ID>18</REF_ID>
      <FORMAT_CODE>NOS</FORMAT_CODE>
      <DESCR_TXT>Canadian Source</DESCR_TXT>
      <SECURITY_CLSFC_CODE>NU</SECURITY_CLSFC_CODE>
      <SOURCE_TXT>Grove Contact Information</SOURCE_TXT>
      <TRANS_TYPE_CODE>EMLMSG</TRANS_TYPE_CODE>
      <OWNER_ID>7</OWNER_ID>
      <UPDATE_SEQNR>1</UPDATE_SEQNR>
    </REF>
  </REF_TBL>
  <OBJ_ITEM_TBL>
    <OBJ_ITEM>
      <OBJ_ITEM_ID>28</OBJ_ITEM_ID>
      <CAT_CODE>MA</CAT_CODE>
      <NAME>Bad Guy 1</NAME>
      <ALTN_IDENTIFIC_TXT>BAD001</ALTN_IDENTIFIC_TXT>
      <OWNER_ID>7</OWNER_ID>
      <UPDATE_SEQNR>1</UPDATE_SEQNR>
    </OBJ_ITEM>
    <OBJ_ITEM>
      <OBJ_ITEM_ID>10</OBJ_ITEM_ID>
      <CAT_CODE>MA</CAT_CODE>
      <NAME>HMCS Grove</NAME>
      <ALTN_IDENTIFIC_TXT>CAD001</ALTN_IDENTIFIC_TXT>
      <OWNER_ID>7</OWNER_ID>
      <UPDATE_SEQNR>1</UPDATE_SEQNR>
    </OBJ_ITEM>
  </OBJ_ITEM_TBL>
  <MAT_TBL>
    <MAT>
      <MAT_ID>28</MAT_ID>
      <SERIAL_NO_ID_TXT>Model 1234</SERIAL_NO_ID_TXT>
      <LOT_IDENTIFIC_TXT></LOT_IDENTIFIC_TXT>
      <BODY_COLOUR_CODE>ORANGE</BODY_COLOUR_CODE>
      <MARKING_CODE>SYMBOL</MARKING_CODE>
      <MARKING_COLOUR_CODE>YELLOW</MARKING_COLOUR_CODE>
      <OWNER_ID>7</OWNER_ID>
      <UPDATE_SEQNR>1</UPDATE_SEQNR>
    </MAT>
  </MAT_TBL>
  <ORG_TBL>
    <ORG>
```

```
      <ORG_ID>28</ORG_ID>
      <CAT_CODE>UN</CAT_CODE>
      <NICKNAME_NAME>Bad Guy</NICKNAME_NAME>
      <OWNER_ID>7</OWNER_ID>
      <UPDATE_SEQNR>1</UPDATE_SEQNR>
    </ORG>
    <ORG>
      <ORG_ID>10</ORG_ID>
      <CAT_CODE>UN</CAT_CODE>
      <NICKNAME_NAME>CF</NICKNAME_NAME>
      <OWNER_ID>7</OWNER_ID>
      <UPDATE_SEQNR>1</UPDATE_SEQNR>
    </ORG>
  </ORG_TBL>
  <OBJ_TYPE_TBL>
    <OBJ_TYPE>
      <OBJ_TYPE_ID>4</OBJ_TYPE_ID>
      <CAT_CODE>MA</CAT_CODE>
      <DUMMY_IND_CODE>YES</DUMMY_IND_CODE>
      <NAME>Sub</NAME>
      <NATIONALITY_CODE>BK</NATIONALITY_CODE>
      <OWNER_ID>7</OWNER_ID>
      <UPDATE_SEQNR>1</UPDATE_SEQNR>
    </OBJ_TYPE>
    <OBJ_TYPE>
      <OBJ_TYPE_ID>3</OBJ_TYPE_ID>
      <CAT_CODE>MA</CAT_CODE>
      <DUMMY_IND_CODE>YES</DUMMY_IND_CODE>
      <NAME>Frigate</NAME>
      <NATIONALITY_CODE>CA</NATIONALITY_CODE>
      <OWNER_ID>7</OWNER_ID>
      <UPDATE_SEQNR>1</UPDATE_SEQNR>
    </OBJ_TYPE>
  </OBJ_TYPE_TBL>
  <MAT_TYPE_TBL>
    <MAT_TYPE>
      <MAT_TYPE_ID>3</MAT_TYPE_ID>
      <CAT_CODE>NOS</CAT_CODE>
      <RPTBL_ITEM_TXT></RPTBL_ITEM_TXT>
      <STOCK_NO_TXT></STOCK_NO_TXT>
      <SUPPLY_CLASS_CODE></SUPPLY_CLASS_CODE>
      <OWNER_ID>7</OWNER_ID>
      <UPDATE_SEQNR>1</UPDATE_SEQNR>
    </MAT_TYPE>
  </MAT_TYPE_TBL>
  <EQPT_TYPE_TBL>
    <EQPT_TYPE>
      <EQPT_TYPE_ID>3</EQPT_TYPE_ID>
      <CAT_CODE>ELCTRN</CAT_CODE>
      <LOADED_WT_QTY></LOADED_WT_QTY>
      <UNLOADED_WT_QTY></UNLOADED_WT_QTY>
      <MAX_HEIGHT_DIM></MAX_HEIGHT_DIM>
      <MAX_LENGTH_DIM></MAX_LENGTH_DIM>
      <MAX_WIDTH_DIM></MAX_WIDTH_DIM>
```

```
    <OWNER_ID>7</OWNER_ID>
    <UPDATE_SEQNR>1</UPDATE_SEQNR>
  </EQPT_TYPE>
</EQPT_TYPE_TBL>
<RPTD_TBL>
  <RPTD>
    <RPTD_ID>8</RPTD_ID>
    <ACCURACY_CODE>6</ACCURACY_CODE>
    <CAT_CODE>REP</CAT_CODE>
    <CNTG_IND_CODE>NO</CNTG_IND_CODE>
    <CREDIBILITY_CODE>RPTUNC</CREDIBILITY_CODE>
    <RELIABILITY_CODE>A</RELIABILITY_CODE>
    <REP_DATE>20031007</REP_DATE>
    <REP_TIME>111000</REP_TIME>
    <SOURCE_TYPE_CODE>CONTAC</SOURCE_TYPE_CODE>
    <TIMING_CAT_CODE>RDABST</TIMING_CAT_CODE>
    <REF_ID>18</REF_ID>
    <REP_ORG_ID>10</REP_ORG_ID>
    <ENT_CAT_CODE>OILOCA</ENT_CAT_CODE>
    <OWNER_ID>7</OWNER_ID>
    <UPDATE_SEQNR>1</UPDATE_SEQNR>
  </RPTD>
</RPTD_TBL>
<RPTD_TBL>
  <RPTD>
    <RPTD_ID>1</RPTD_ID>
    <ACCURACY_CODE></ACCURACY_CODE>
    <CAT_CODE>REP</CAT_CODE>
    <CNTG_IND_CODE></CNTG_IND_CODE>
    <CREDIBILITY_CODE></CREDIBILITY_CODE>
    <RELIABILITY_CODE></RELIABILITY_CODE>
    <REP_DATE>20031007</REP_DATE>
    <REP_TIME></REP_TIME>
    <SOURCE_TYPE_CODE></SOURCE_TYPE_CODE>
    <TIMING_CAT_CODE>RDABST</TIMING_CAT_CODE>
    <REF_ID>18</REF_ID>
    <REP_ORG_ID>10</REP_ORG_ID>
    <ENT_CAT_CODE>OILOCA</ENT_CAT_CODE>
    <OWNER_ID>7</OWNER_ID>
    <UPDATE_SEQNR>1</UPDATE_SEQNR>
  </RPTD>
  <RPTD>
    <RPTD_ID>4</RPTD_ID>
    <ACCURACY_CODE></ACCURACY_CODE>
    <CAT_CODE>REP</CAT_CODE>
    <CNTG_IND_CODE></CNTG_IND_CODE>
    <CREDIBILITY_CODE></CREDIBILITY_CODE>
    <RELIABILITY_CODE></RELIABILITY_CODE>
    <REP_DATE>20031007</REP_DATE>
    <REP_TIME></REP_TIME>
    <SOURCE_TYPE_CODE></SOURCE_TYPE_CODE>
    <TIMING_CAT_CODE>RDABST</TIMING_CAT_CODE>
    <REF_ID>18</REF_ID>
    <REP_ORG_ID>10</REP_ORG_ID>
```

```
      <ENT_CAT_CODE>OILOCA</ENT_CAT_CODE>
      <OWNER_ID>7</OWNER_ID>
      <UPDATE_SEQNR>1</UPDATE_SEQNR>
    </RPTD>
  </RPTD_TBL>
  <RPTD_ABS_TIMING_TBL>
    <RPTD_ABS_TIMING>
      <RPTD_ABS_TIMING_RPTD_ID>8</RPTD_ABS_TIMING_RPTD_ID>
      <DUR>3600</DUR>
      <EFFCTV_DATE>20030623</EFFCTV_DATE>
      <EFFCTV_TIME>111000</EFFCTV_TIME>
      <EFFCTV_TIME_PRECISION_CODE>SECOND</EFFCTV_TIME_PRECISION_CODE>
      <OWNER_ID>7</OWNER_ID>
      <UPDATE_SEQNR>1</UPDATE_SEQNR>
    </RPTD_ABS_TIMING>
  </RPTD_ABS_TIMING_TBL>
  <OBJ_ITEM_STAT_TBL>
    <OBJ_ITEM_STAT>
      <OBJ_ITEM_ID>28</OBJ_ITEM_ID>
      <OBJ_ITEM_STAT_IX>277</OBJ_ITEM_STAT_IX>
      <CAT_CODE>MA</CAT_CODE>
      <HSTLY_CODE>HO</HSTLY_CODE>
      <BOOBY_TRAP_IND_CODE>NO</BOOBY_TRAP_IND_CODE>
      <RPTD_ID>8</RPTD_ID>
      <OWNER_ID>7</OWNER_ID>
      <UPDATE_SEQNR>1</UPDATE_SEQNR>
    </OBJ_ITEM_STAT>
  </OBJ_ITEM_STAT_TBL>
  <CONTXT_TBL>
    <CONTXT>
      <CONTXT_ID>1</CONTXT_ID>
      <NAME>Composite of Tracks 1 and 4</NAME>
      <OWNER_ID>7</OWNER_ID>
      <UPDATE_SEQNR>1</UPDATE_SEQNR>
    </CONTXT>
  </CONTXT_TBL>
  <LOC_TBL>
    <LOC>
      <LOC_ID>5</LOC_ID>
      <CAT_CODE>PT</CAT_CODE>
      <OWNER_ID>7</OWNER_ID>
      <UPDATE_SEQNR>1</UPDATE_SEQNR>
    </LOC>
    <LOC>
      <LOC_ID>6</LOC_ID>
      <CAT_CODE>SURFAC</CAT_CODE>
      <OWNER_ID>7</OWNER_ID>
      <UPDATE_SEQNR>1</UPDATE_SEQNR>
    </LOC>
  </LOC_TBL>
  <LOC_TBL>
    <LOC>
      <LOC_ID>83</LOC_ID>
      <CAT_CODE>VL</CAT_CODE>
```

```
      <OWNER_ID>7</OWNER_ID>
      <UPDATE_SEQNR>1</UPDATE_SEQNR>
    </LOC>
  </LOC_TBL>
  <POINT_TBL>
    <POINT>
      <POINT_ID>5</POINT_ID>
      <CAT_CODE>ABS</CAT_CODE>
      <OWNER_ID>7</OWNER_ID>
      <UPDATE_SEQNR>1</UPDATE_SEQNR>
    </POINT>
  </POINT_TBL>
  <VER_DIST_TBL>
    <VER_DIST>
      <VER_DIST_ID>81</VER_DIST_ID>
      <CAT_CODE>HEIGHT</CAT_CODE>
      <DIM>110</DIM>
      <PRECISION_CODE>1M</PRECISION_CODE>
      <OWNER_ID>7</OWNER_ID>
      <UPDATE_SEQNR>1</UPDATE_SEQNR>
    </VER_DIST>
    <VER_DIST>
      <VER_DIST_ID>82</VER_DIST_ID>
      <CAT_CODE>HEIGHT</CAT_CODE>
      <DIM>90</DIM>
      <PRECISION_CODE>1M</PRECISION_CODE>
      <OWNER_ID>7</OWNER_ID>
      <UPDATE_SEQNR>1</UPDATE_SEQNR>
    </VER_DIST>
  </VER_DIST_TBL>
  <ABS_POINT_TBL>
    <ABS_POINT>
      <ABS_POINT_ID>5</ABS_POINT_ID>
      <LAT_COORD>45</LAT_COORD>
      <LONG_COORD>-50</LONG_COORD>
      <ANGULAR_PRECISION_CODE>MINUTE</ANGULAR_PRECISION_CODE>
      <ABS_POINT_VER_DIST_ID></ABS_POINT_VER_DIST_ID>
      <OWNER_ID>7</OWNER_ID>
      <UPDATE_SEQNR>1</UPDATE_SEQNR>
    </ABS_POINT>
  </ABS_POINT_TBL>
  <OBJ_ITEM_LOC_TBL>
    <OBJ_ITEM_LOC>
      <LOC_ID>83</LOC_ID>
      <OBJ_ITEM_ID>28</OBJ_ITEM_ID>
      <OBJ_ITEM_LOC_IX>1</OBJ_ITEM_LOC_IX>
      <ACCURACY_QTY>10</ACCURACY_QTY>
      <BEARING_ANGLE>90</BEARING_ANGLE>
      <SPEED_RATE>18</SPEED_RATE>
      <GEOMETRY_CFEAT_TYPE_ID></GEOMETRY_CFEAT_TYPE_ID>
      <RPTD_ID>8</RPTD_ID>
      <OWNER_ID>7</OWNER_ID>
      <UPDATE_SEQNR>1</UPDATE_SEQNR>
    </OBJ_ITEM_LOC>
```

```
    <OBJ_ITEM_LOC>
      <LOC_ID>5</LOC_ID>
      <OBJ_ITEM_ID>10</OBJ_ITEM_ID>
      <OBJ_ITEM_LOC_IX>13</OBJ_ITEM_LOC_IX>
      <ACCURACY_QTY>10</ACCURACY_QTY>
      <BEARING_ANGLE></BEARING_ANGLE>
      <SPEED_RATE></SPEED_RATE>
      <GEOMETRY_CFEAT_TYPE_ID></GEOMETRY_CFEAT_TYPE_ID>
      <RPTD_ID>8</RPTD_ID>
      <OWNER_ID>7</OWNER_ID>
      <UPDATE_SEQNR>1</UPDATE_SEQNR>
    </OBJ_ITEM_LOC>
  </OBJ_ITEM_LOC_TBL>
  <SURFACE_TBL>
    <SURFACE>
      <SURFACE_ID>6</SURFACE_ID>
      <CAT_CODE>FA</CAT_CODE>
      <OWNER_ID>7</OWNER_ID>
      <UPDATE_SEQNR>1</UPDATE_SEQNR>
    </SURFACE>
  </SURFACE_TBL>
  <FAN_AREA_TBL>
    <FAN_AREA>
      <FAN_AREA_ID>6</FAN_AREA_ID>
      <MNM_RANGE_DIM>11.289</MNM_RANGE_DIM>
      <MAX_RANGE_DIM>12.711</MAX_RANGE_DIM>
      <ORIENT_ANGLE>263.25</ORIENT_ANGLE>
      <SECTOR_SIZE_ANGLE>13.5</SECTOR_SIZE_ANGLE>
      <FAN_AREA_VERTEX_POINT_ID>5</FAN_AREA_VERTEX_POINT_ID>
      <OWNER_ID>7</OWNER_ID>
      <UPDATE_SEQNR>1</UPDATE_SEQNR>
    </FAN_AREA>
  </FAN_AREA_TBL>
  <CONTXT_RPTD_ASSOC_TBL>
    <CONTXT_RPTD_ASSOC>
      <CONTXT_ID>1</CONTXT_ID>
      <RPTD_ID>8</RPTD_ID>
      <CONTXT_RPTD_ASSOC_IX>1</CONTXT_RPTD_ASSOC_IX>
      <CAT_CODE>IMPL</CAT_CODE>
      <OWNER_ID>7</OWNER_ID>
      <UPDATE_SEQNR>1</UPDATE_SEQNR>
    </CONTXT_RPTD_ASSOC>
  </CONTXT_RPTD_ASSOC_TBL>
  <ELCTRNC_EQPT_TYPE_TBL>
    <ELCTRNC_EQPT_TYPE>
      <ELCTRNC_EQPT_TYPE_ID>3</ELCTRNC_EQPT_TYPE_ID>
      <CAT_CODE>SEN</CAT_CODE>
      <SUBCAT_CODE>SONAR</SUBCAT_CODE>
      <OWNER_ID>7</OWNER_ID>
      <UPDATE_SEQNR>1</UPDATE_SEQNR>
    </ELCTRNC_EQPT_TYPE>
  </ELCTRNC_EQPT_TYPE_TBL>
  <CAPAB_TBL>
    <CAPAB>
```

```
      &lt;CAPAB_ID&gt;77&lt;/CAPAB_ID&gt;
      &lt;CAT_CODE&gt;MOBL&lt;/CAT_CODE&gt;
      &lt;SUBCAT_CODE&gt;&lt;/SUBCAT_CODE&gt;
      &lt;DAY_NIGHT_CODE&gt;DN&lt;/DAY_NIGHT_CODE&gt;
      &lt;UOM_CODE&gt;EA&lt;/UOM_CODE&gt;
      &lt;OWNER_ID&gt;7&lt;/OWNER_ID&gt;
      &lt;UPDATE_SEQNR&gt;1&lt;/UPDATE_SEQNR&gt;
    &lt;/CAPAB&gt;
  &lt;/CAPAB_TBL&gt;
  &lt;MOB_CAPAB_TBL&gt;
    &lt;MOB_CAPAB&gt;
      &lt;MOB_CAPAB_ID&gt;77&lt;/MOB_CAPAB_ID&gt;
      &lt;CAT_CODE&gt;SEASS&lt;/CAT_CODE&gt;
      &lt;TERRAIN_TYPE_CODE&gt;NOS&lt;/TERRAIN_TYPE_CODE&gt;
      &lt;OWNER_ID&gt;7&lt;/OWNER_ID&gt;
      &lt;UPDATE_SEQNR&gt;1&lt;/UPDATE_SEQNR&gt;
    &lt;/MOB_CAPAB&gt;
  &lt;/MOB_CAPAB_TBL&gt;
  &lt;OBJ_ITEM_CAPAB_TBL&gt;
    &lt;OBJ_ITEM_CAPAB&gt;
      &lt;OBJ_ITEM_ID&gt;28&lt;/OBJ_ITEM_ID&gt;
      &lt;CAPAB_ID&gt;77&lt;/CAPAB_ID&gt;
      &lt;OBJ_ITEM_CAPAB_IX&gt;23&lt;/OBJ_ITEM_CAPAB_IX&gt;
      &lt;MSN_PRIMACY_CODE&gt;PRIME&lt;/MSN_PRIMACY_CODE&gt;
      &lt;QTY&gt;1&lt;/QTY&gt;
      &lt;RPTD_ID&gt;8&lt;/RPTD_ID&gt;
      &lt;OWNER_ID&gt;7&lt;/OWNER_ID&gt;
      &lt;UPDATE_SEQNR&gt;1&lt;/UPDATE_SEQNR&gt;
    &lt;/OBJ_ITEM_CAPAB&gt;
  &lt;/OBJ_ITEM_CAPAB_TBL&gt;
  &lt;CONTXT_ELMT_TBL&gt;
    &lt;CONTXT_ELMT&gt;
      &lt;CONTXT_ID&gt;1&lt;/CONTXT_ID&gt;
      &lt;CONTXT_ELMT_IX&gt;1&lt;/CONTXT_ELMT_IX&gt;
      &lt;RPTD_ID&gt;1&lt;/RPTD_ID&gt;
      &lt;OWNER_ID&gt;7&lt;/OWNER_ID&gt;
      &lt;UPDATE_SEQNR&gt;1&lt;/UPDATE_SEQNR&gt;
    &lt;/CONTXT_ELMT&gt;
    &lt;CONTXT_ELMT&gt;
      &lt;CONTXT_ID&gt;1&lt;/CONTXT_ID&gt;
      &lt;CONTXT_ELMT_IX&gt;2&lt;/CONTXT_ELMT_IX&gt;
      &lt;RPTD_ID&gt;4&lt;/RPTD_ID&gt;
      &lt;OWNER_ID&gt;7&lt;/OWNER_ID&gt;
      &lt;UPDATE_SEQNR&gt;1&lt;/UPDATE_SEQNR&gt;
    &lt;/CONTXT_ELMT&gt;
  &lt;/CONTXT_ELMT_TBL&gt;
  &lt;GEOM_VOL_TBL&gt;
    &lt;GEOM_VOL&gt;
      &lt;GEOM_VOL_ID&gt;83&lt;/GEOM_VOL_ID&gt;
      &lt;CAT_CODE&gt;SURVOL&lt;/CAT_CODE&gt;
      &lt;GEOM_VOL_LOWER_VER_DIST_ID&gt;82&lt;/GEOM_VOL_LOWER_VER_DIST_ID&gt;
      &lt;GEOM_VOL_UPPER_VER_DIST_ID&gt;81&lt;/GEOM_VOL_UPPER_VER_DIST_ID&gt;
      &lt;OWNER_ID&gt;7&lt;/OWNER_ID&gt;
      &lt;UPDATE_SEQNR&gt;1&lt;/UPDATE_SEQNR&gt;
```

```
    </GEOM_VOL>
  </GEOM_VOL_TBL>
  <SURFACE_VOL_TBL>
    <SURFACE_VOL>
      <SURFACE_VOL_ID>83</SURFACE_VOL_ID>
      <OWNER_ID>7</OWNER_ID>
      <UPDATE_SEQNR>1</UPDATE_SEQNR>
      <SURFACE_VOL_DFNG_SURFACE_ID>6</SURFACE_VOL_DFNG_SURFACE_ID>
    </SURFACE_VOL>
  </SURFACE_VOL_TBL>
</GH5Complete>
```

List of symbols/abbreviations/acronyms/initialisms

ARM	Automated Replication Mechanism
ASF	Apache Software Foundation
ATCCIS	Army Tactical Command and Control Information System
CDS	Coalition Data Server
CF	Canadian Forces
DND	Department of National Defence
DRDC Atlantic	Defence R&D Canada – Atlantic
FK	Foreign Key
GH	Generic Hub
GUI	Graphical User Interface
HTTP	Hyper Text Transfer Protocol
IDEF1X	Integration Definition for Information Modelling
JAR	Java Archive
JSP	Java Server Pages
LC2IEDM	Land Command and Control Information Exchange Data Model
LC2IEDB	Land Command and Control Information Exchange Database
MAR	Maritime Systems
MIST	Maritime Information Sharing Technology
NATO	North Atlantic Treaty Organisation
NUWC	Naval Undersea Warfare Center
OCXS	Operational Context Exchange Service

RPC	Remote Procedure Call
SOAP	Simple Object Access Protocol
SQL	Structured Query Language
TP	Technical Panel
TTCP	The Technical Cooperation Program
UK	United Kingdom
URI	Universal Resource Identifier
UTF	Unicode Transformation Format
US	United States
W3C	World Wide Web Consortium
WEBDAV	Web-based Distributed Authoring and Versioning
XML	eXtensible Markup Language
XSLT	eXtensible Stylesheet Language Transformations

Distribution list

LIST PART 1:

2 DRDC ATLANTIC LIBRARY FILE COPIES
3 DRDC ATLANTIC LIBRARY (SPARES)
1 W. A. ROGER
1 M. E. LEFRANÇOIS
1 G. R. MELLEMA
1 J. S. KENNEDY
1 AUTHOR

10 TOTAL LIST PART 1

LIST PART 2: DISTRIBUTED BY AUTHOR

1 Frederick G. Burkley
Code 2212, Bldg. 1171-1
Naval Undersea Warfare Center
Middletown, RI, USA 02841

1 Erik Chaum
TTCP MAR TP-1 USNL
Naval Undersea Warfare Center
Division Newport
Code 2231, Bldg 1171-3
1176 Howell Street
Newport, RI 02841-1708

1 Eric Dorion
Metrics & Experimentation Group
System of Systems Section
Defence R&D Canada - Valcartier
2459 Pie-XI Blvd. North
Val-Belair, Quebec
Canada, G3J 1X5

1 LCol Jacques Hamel, CD, Ing
Project Manager
Intelligence, Surveillance, Target Acquisition and Reconnaissance (PM ISTAR)
National Defence Headquarters
MGen George R. Pearkes
101 Colonel By Drive
Ottawa, Ontario Canada
K1A 0K2

1 Richard King
TTCP MAR TP-1 Chair
Dstl Naval Systems Dept.
Room C134
Portsdown West
Fareham
Hampshire PO17 6AD, UK

1 Mr. Mark Lancester
QinetiQ
Building A22,
Winfrith Technology Centre,
Winfrith Newburgh,
Dorchester,
Dorset.
DT2 8XJ
United Kingdom

1 Shane Lunga
6 Mercury Building,
Dstl Portsdown West,
Fareham, PO17 6AD,
United Kingdom

1 Graham MacFerson
TTCP MAR TP-1 NZNL
Section Manager, C4I & Simulation
Defence Technology Agency
Private Bag 32901
Auckland, New Zealand

1 Roger Manning
DSTO Edinburgh
Maritime Operations Division
PO Box 1500
Salisbury
SA 5108
Australia

1 James Oblinger
Naval Undersea Warfare Center Division Newport
Code 2233, Bldg 1171-2
Newport, RI 02841

1 Dr. John Riley
TTCP MAR TP-1 ASNL
Defence Science and Technology Organisation
Maritime Operations Division
PO Box 1500, Edinburgh, SA 5111
Australia

1 James Taylor
Naval Undersea Warfare Center
Division Newport
Code 2231 Bldg. 1171-3
1176 Howell St
Newport RI 02841

1 Robert Walden
TTCP MAR TP-1 UKNL
Naval Systems Department, C-134/EC
Defence Science and Technology Laboratory,
Portsdown West, Fareham, Hampshire,
UK P017 6AD

13 TOTAL LIST PART 2

LIST PART 3: DISTRIBUTED BY DRDKIM 3

1 NDHQ/ ADMS&T/ DRDKIM 3

1 TOTAL LIST PART 3

24 TOTAL COPIES REQUIRED

DOCUMENT CONTROL DATA

(Security classification of title, body of abstract and indexing annotation must be entered when the overall document is classified)

1. ORIGINATOR (the name and address of the organization preparing the document. Organizations for whom the document was prepared, e.g. Centre sponsoring a contractor's report, or tasking agency, are entered in section 8.) Defence R&D Canada – Atlantic PO Box 1012 Dartmouth, Nova Scotia B2Y 3Z7	2. SECURITY CLASSIFICATION (overall security classification of the document including special warning terms if applicable). UNCLASSIFIED

3. TITLE (the complete document title as indicated on the title page. Its classification should be indicated by the appropriate abbreviation (S,C,R or U) in parentheses after the title).

Placing Tactical Data into the MIST and LC2IEDM Systems

4. AUTHORS (Last name, first name, middle initial. If military, show rank, e.g. Doe, Maj. John E.)

Anthony W. Isenor

5. DATE OF PUBLICATION (month and year of publication of document) October 2003	6a. NO. OF PAGES (total containing information Include Annexes, Appendices, etc). 80	6b. NO. OF REFS (total cited in document) 25

7. DESCRIPTIVE NOTES (the category of the document, e.g. technical report, technical note or memorandum. If appropriate, enter the type of report, e.g. interim, progress, summary, annual or final. Give the inclusive dates when a specific reporting period is covered).

TECHNICAL MEMORANDUM

8. SPONSORING ACTIVITY (the name of the department project office or laboratory sponsoring the research and development. Include address).

Defence R&D Canada – Atlantic
PO Box 1012
Dartmouth, NS, Canada B2Y 3Z7

9a. PROJECT OR GRANT NO. (if appropriate, the applicable research and development project or grant number under which the document was written. Please specify whether project or grant). 11cs.11	9b. CONTRACT NO. (if appropriate, the applicable number under which the document was written).
10a ORIGINATOR'S DOCUMENT NUMBER (the official document number by which the document is identified by the originating activity. This number must be unique to this document.) TM 2003 - 168	10b OTHER DOCUMENT NOs. (Any other numbers which may be assigned this document either by the originator or by the sponsor.) DRDC Atlantic

11. DOCUMENT AVAILABILITY (any limitations on further dissemination of the document, other than those imposed by security classification)

(x) Unlimited distribution
(   ) Defence departments and defence contractors; further distribution only as approved
(   ) Defence departments and Canadian defence contractors; further distribution only as approved
(   ) Government departments and agencies; further distribution only as approved
(   ) Defence departments; further distribution only as approved
(   ) Other (please specify):

12. DOCUMENT ANNOUNCEMENT (any limitation to the bibliographic announcement of this document. This will normally correspond to the Document Availability (11). However, where further distribution (beyond the audience specified in (11) is possible, a wider announcement audience may be selected).

13. ABSTRACT (a brief and factual summary of the document. It may also appear elsewhere in the body of the document itself. It is highly desirable that the abstract of classified documents be unclassified. Each paragraph of the abstract shall begin with an indication of the security classification of the information in the paragraph (unless the document itself is unclassified) represented as (S), (C), (R), or (U). It is not necessary to include here abstracts in both official languages unless the text is bilingual).

This document presents a review of the Maritime Information Sharing Technology (MIST) database and the Land Command and Control Information Exchange Data Model (LC2IEDM) in the context of storing sonar contact information. The set-up of each system is described, as well as the software that supports the systems. Tactical data describing a sonar contact is defined and used to load the MIST database. The same data content is then placed into appropriate tables in the Land Command and Control Information Exchange Database (LC2IEDB). Obvious strengths and weaknesses of both systems are described.

14. KEYWORDS, DESCRIPTORS or IDENTIFIERS (technically meaningful terms or short phrases that characterize a document and could be helpful in cataloguing the document. They should be selected so that no security classification is required. Identifiers, such as equipment model designation, trade name, military project code name, geographic location may also be included. If possible keywords should be selected from a published thesaurus. e.g. Thesaurus of Engineering and Scientific Terms (TEST) and that thesaurus-identified. If it not possible to select indexing terms which are Unclassified, the classification of each should be indicated as with the title).

Maritime Information Sharing Technology
MIST
Coalition Data Server
CDS
Generic Hub
GH
Land Command and Control Information Exchange Data Model
LC2IEDM
Operational Context Exchange Service
OCXS
system of systems
Networked Underwater Warfare

This page intentionally left blank.

Defence R&D Canada

Canada's leader in defence
and national security R&D

R & D pour la défense Canada

Chef de file au Canada en R & D
pour la défense et la sécurité nationale

DEFENCE R&D DÉFENSE

www.drdc-rddc.gc.ca